BAYESIAN APPROACH FOR FEASIBILITY DETERMINATION AND SPATIOTEMPORAL SCHEDULING

A Dissertation
Presented to
The Academic Faculty

By

Junying He

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Engineering
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology

August 2020

BAYESIAN APPROACH FOR FEASIBILITY DETERMINATION AND SPATIOTEMPORAL SCHEDULING

Thesis committee:

Dr. Seong-Hee Kim
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yajun Mei
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Sigrún Andradóttir
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Enlu Zhou
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Chuljin Park
Department of Industrial Engineering
Hanyang University

Date approved: June 10, 2020

Stay hungry. Stay foolish.

Steve Jobs

For all my loved ones

ACKNOWLEDGMENTS

The five-year PhD study at Georgia Tech is definitely one of the most memorable times in my life for which I will forever be grateful. I feel fortunate to have reached this point and I owe thanks to all the people who have offered me tremendous help along this journey.

First of all, I would like to express my deepest gratitude to my advisor, Dr. Seong-Hee Kim, for her exceptional guidance, support and patience at all times. I still remember that the second year of my PhD study was a very difficult time, as my previous advisor decided not to work with me and my first attempt to the qualification exam was failed. I tried to look for an advisor and that was the first time I met Dr. Kim. I'm extremely grateful that Dr. Kim agreed to be my advisor and lead me into the study of simulation. If it weren't for her generous support, I could have dropped out of the PhD program. Since then, Dr. Kim and I have been working on several research projects and I am deeply impressed by her intelligence and expertise. Every meeting with her has been very inspiring and helpful. Without her contribution, this thesis could not be completed. I appreciate that Dr. Kim has always been very considerate to me, as she is caring for any concern that I may have in my life. Also I very much enjoyed the meals and coffees that she bought for me.

I am grateful to my other committee members: Dr. Sigrún Andradóttir, Dr. Yajun Mei, Dr. Enlu Zhou, and Dr. Chuljin Park, for their willingness to serve in my committee. I would like to thank for their time and effort in providing valuable input to make this thesis more complete.

I am grateful to all the faculty and staff members at Georgia Tech for teaching me valuable knowledge and giving me a comfortable school life. I want to give special thanks to Dr. Yajun Mei and Dr. Yao Xie for their support in my admission to the program.

I would also like to thank all my friends and colleagues for bringing so much happiness into my life. Especially, I want to thank Yujie Guo and Sai Vasam for being my fond roommates. I would like to thank Chen Feng, Xi He and Chuanping Yu for being my mates

ever since the first year of the program.

At last, I would like to give special thanks to my parents, Cuiping Liu and Xingxiong He, for their endless love and encouragement. For almost a decade, I have been on the other side of the world from my parents and missed a lot of opportunities of companion with them. However, my parents have always been very understanding and supportive for all my decisions. Over all these years, my parents have always been my strongest backing. Any of my accomplishment could not have been done without their selfless love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis mainly consists of four parts. The first three parts explore Bayesian methods in solving feasibility determination problems that commonly arise in the study of simulation and the last part considers spatiotemporal scheduling in manufacturing. More specifically, we propose a new reward function for Bayesian feasibility determination, which emphasizes the importance of barely feasible/infeasible systems whose mean performance measures are close to the threshold. We utilize our proposed reward function in developing a Bayesian procedure and show its advantage in comparison with the benchmark procedures. Then, we present new two-stage and sequential Bayesian procedures that are not only easy to compute but also effective in solving the feasibility determination problem. In the third part, we focus on solving feasibility determination using a Gaussian process and propose a Bayesian procedure with our novel acquisition function. Finally, the last part focuses on a different topic, namely, spatiotemporal scheduling, which often occurs in a manufacturing site where products are large and tend to be customized, such as ships, aircraft, and constructional structures. We consider how to generate a reasonably good temporal and spatial schedule on the manufacturing process. We propose a two-phase approach in solving this scheduling problem with an application to block assembly process in shipbuilding.

CHAPTER 1
INTRODUCTION

For a complex industrial system, it is often very difficult for the decision maker to analytically identify the best or a good alternative for its manufacturing or service process, especially when there are numerous possible alternatives under a stochastic environment. As computing power grows stronger nowadays, computing-aided methods such as simulation and optimization programs are frequently adopted as decision supporting tools in a wide range of applications. In the following, we introduce the feasibility determination problem and spatiotemporal scheduling problem, which are the two main focuses of this thesis. Then, we make more discussion on recent developments that utilize simulation and optimization programs in solving these decision making problems.

In general, a problem of feasibility determination is to find a set of feasible alternatives among a finite number of simulated alternatives in the presence of stochastic constraints. If a constraint is imposed on a performance measure whose value can only be estimated by stochastic simulation, we call it a stochastic constraint. Feasibility determination for a stochastic constraint occurs in many management and industrial applications. Some real-world examples are as follows:

1. The emergency department of a health-care unit has several shift arrangements for the staff. The decision maker wants to know which arrangements can keep patients' mean waiting time no more than 30 minutes.

2. A manufacturing company has a few available production plans, and the decision maker wants to identify which plans can satisfy the production amount requirement of 10,000 units per month.

3. A facility management team is considering a number of cooling options on hand and

wants to find out which options can keep the facility's temperature lower than or equal to $85°$F.

The problem of feasibility determination originates from a broader class of problems called ranking-and-selection (R&S). R&S problems are to compare a finite number of simulated alternatives. A typical R&S problem is choosing the best system among a finite number of simulated systems, where the best system is defined as the one with the largest or smallest expected performance measure. This selection problem is called selection of the best. Famous R&S problems also include comparison with a standard, bernoulli selection and binomial selection. For a general overview of the selection of the best problem and other types of R&S problems, one can refer to [1, 2, 3].

A spatiotemporal scheduling problem often occurs in a manufacturing site where products are large and tend to be customized such as ships, airplanes and construction structures. These products are usually produced by first assembling their basic components and then putting them together in the final production. Each basic component is often called a block. During the production process, there are a large number of blocks to be assembled inside numerous workshops called bays. In a spatiotemporal scheduling problem, the goal is usually to find a feasible schedule that gives information on when and where to assemble each block, while optimizing the efficiency of the production process. In other words, it is necessary to perform both temporal and spatial scheduling for the block assembly process that satisfies a set of constraints.

In this chapter, we review R&S literature in Section 1.1 and provide an overview of this thesis on the feasibility determination problem in Section 1.2. Then Section 1.3 gives an introduction of a spatiotemporal scheduling problem and discusses relevant literature.

1.1 Ranking-and-Selection

The classical selection of the best problem is on a single performance measure, such as finding an inventory policy with the smallest expected cost. The existing procedures for the

selection of the best problem adopt one of the three general frameworks: the indifference-zone (IZ) approach, the optimal computing budget allocation (OCBA) approach, and the Bayesian approach. An IZ procedure usually focuses on providing a guarantee for the probability of a correct selection (PCS), where the mean performance of the best system is assumed to be at least a user-specified constant better than other alternatives. This user-specified constant is called the indifference-zone (IZ) parameter and interpreted as the minimum difference worth detection. [4, 5] provide good surveys on IZ procedures for selection of the best. Some useful IZ procedures include [6, 7, 8, 9]. While IZ procedures try to decide how many observations to take in order to provide a guarantee on PCS, procedures with the OCBA approach allocate a finite computing budget sequentially in order to maximize PCS. General references of the OCBA approach can be found in [10, 11]. [12] presents one of the earliest OCBA procedures for selection of the best, and more follow-up procedures have been developed in [13, 14, 15, 16]. Finally, a Bayesian procedure tries to find a sampling strategy to minimize the Bayesian loss or maximize the Bayesian reward. It assumes the unknown parameters of a sampling distribution are random variables and uses the Bayesian rule to infer their values. [17] provides a detailed introduction to the Bayesian methodology in R&S. Literature on Bayesian procedures for selection of the best includes [18, 19, 20, 21, 22, 23, 24]. For comparisons among the three approaches, one can refer to [25].

The literature concerning constrained R&S problems is relatively new. Constrained R&S for selection of the best is to find a system with the largest or smallest expected performance measure while satisfying a set of stochastic constraints. For example, one may want to find an inventory policy with the smallest expected cost while the fulfillment rate is greater than or equal to a constant threshold. The constraint on the fulfillment rate is stochastic in a sense that the rate needs to be estimated by simulation. Thus, constrained R&S for the selection of the best problem usually requires statistical procedures for two different goals: feasibility determination and selection of the best feasible system, where the

feasibility determination problem is to identify feasible alternatives that satisfy the stochastic constraints.

Similar to unconstrained R&S, the same three approaches are often used for constrained R&S problems. Based on the IZ approach, [26] considers a general form of a single stochastic constraint on a secondary performance measure and provides procedures for both feasibility determination and selection of the best feasible system. They introduce a parameter called *error tolerance* for feasibility determination, which is similar to the IZ parameter in the selection of the best problem. The error tolerance specifies how much a system's mean performance measure could be off from the threshold in the constraint but still acceptable to the decision maker. [27, 28] focus on feasibility determination and provide IZ procedures for identifying a set of feasible systems in the presence of multiple stochastic constraints. Later, [29, 30] present IZ procedures with slightly tighter bounds for feasibility determination and combine them with procedures to select the best feasible system. Among procedures with the OCBA approach, [31] proposes a budget allocation rule to maximize PCS under multiple stochastic constraints. Instead of selecting a single best feasible system as in [31], [32] develops an OCBA procedure that returns a set of feasible systems in the presence of multiple stochastic constraints. For the Bayesian approach, [33] present a sampling procedure that compares multiple systems to a known standard on a single performance measure. This is essentially feasibility determination under a single stochastic constraint, because a system that has a mean performance measure better than the standard is equivalent to a feasible system. There are additional procedures that use the large deviation principle. While [34, 35] focus on the selection of the best feasible system, [36, 37] provide procedures to find a set of feasible systems. For feasibility determination, [32] also uses the large deviation technique within the OCBA framework.

As mentioned above, procedures for constrained R&S usually have two goals: feasibility determination and selection of the best. To find the best feasible system, one has to consider feasibility determination. The first main focus of this thesis is on feasibility

determination, and thus we discuss more details about the three approaches for feasibility determination.

The IZ procedures for feasibility determination such as [26, 28, 30] are easy to implement, suitable for unequal sampling variances, and they can provide a statistical guarantee on the probability of a correct decision (PCD) on feasibility. However, when there are multiple constraints, it can be inconvenient for the decision maker to choose an error tolerance for each constraint. Moreover, the procedures are designed under the slippage configuration, which assumes that the mean performance measure corresponding to a constraint is exactly the error tolerance away from the constraint threshold. As a result, when the mean performance measure is far away from the threshold in the constraint, the IZ procedures become conservative. Moreover, the IZ procedures try to make a correct decision on feasibility of all constraints while, in fact, we only need to make a correct decision on one of infeasible constraints for a system with one or more infeasible constraints. This also leads to conservative procedures as they usually take more than necessary observations and achieve PCD higher than user-specified, especially when the total number of systems and number of constraints are large. Therefore, the IZ procedures are more suitable in the case that sampling cost is small and the statistical guarantee on feasibility decision is important.

The OCBA procedure for feasibility determination developed in [32] provides asymptotically optimal budget allocation rules to maximize PCD as total budget goes to infinity. However, their allocation rules are derived under the assumptions that sample means and variance of simulation observations are equal to true means and true variances, respectively, for all systems. In addition, the procedure does not maximize PCD directly, but tries to maximize an approximated PCD, which is constructed by the large deviation principle. Another main limitation of this approach is that other than some specific distributions, the large deviation rate functions are generally unknown and need to be estimated. The estimation of the rate function is quite difficult. Thus, issues related to the computational complexity, the decision on initial sample size, and the validation of estimation may arise

for the estimation of the rate function.

Bayesian procedures for feasibility determination in [33] make sampling decisions by considering the value of information, which is defined as the expected reward/loss obtained with an additional sample less the expected reward/loss obtained without that sample. Unlike IZ and OCBA procedures, Bayesian procedures can incorporate the knowledge of sampling cost in the decision making process. The idea is intuitive: one should keep sampling until the value of information from getting one additional sample is not worth its cost. However, the Bayesian procedures are, in general, known to involve heavy calculation, because the calculation of value of information requires taking an expectation over all possible outcomes of the additional observations which have not been taken yet and its closed-form expression is often complicated. Nonetheless, the Bayesian approach has several advantages that explain why our work focuses on Bayesian procedures:

- It has a flexible framework that works with both a unit cost and a total computing budget.
- It treats each observation as expensive so that Bayesian procedures tend to be frugal and help saving sampling cost. This is useful especially for large-scale environmental or physics models, where each simulation run can take long time and high cost.
- Depending on applications, different reward/loss functions can be used to better reflect their objectives. For example, if one is interested in minimizing expected opportunity cost, which is measured by the difference between a system's mean performance and threshold, then a linear loss function can be applied to formulate the problem.

1.2 Feasibility Determination

In this thesis, we first propose a new form of reward function that is more reasonable than the current popular linear and 0-1 reward functions for feasibility determination. Here,

we define systems whose mean performance measures are close to the threshold as barely feasible/infeasible systems, and those far from the threshold as clearly feasible/infeasible systems. The linear functions tend to assign more reward to clearly feasible/infeasible systems than barely feasible/infeasible systems, and the 0-1 functions assign same reward to all systems. Therefore, the linear functions regard the clearly feasible/infeasible systems as more important than the barely feasible/infeasible ones, while the 0-1 functions treat them all equally. However, in many management or operation problems, barely feasible/infeasible systems are often more important. For example, a company's executives may want to find an operational plan that maximizes the expected revenue, while keeping the operating cost smaller than or equal to an allowed amount. Since the company's revenue and operating cost usually move in the opposite directions, the best plan is likely one of those that have operating costs close to the constraint threshold. Thus, it makes sense to put more reward on the barely feasible/infeasible systems than the clearly feasible/infeasible ones. We then apply our reward function in a Bayesian procedure based on work from [33]. [33] presents a framework to find Bayesian optimal policies and uses linear and 0-1 reward functions. Their procedures are fully sequential in a sense that it continues sampling from each system until its posterior mean exits a continuation region for a single constraint. We incorporate our new reward function into this framework of finding the Bayesian optimal policy to show advantages of the new reward function.

Second, we present new two-stage and sequential Bayesian procedures for feasibility determination that are easy to compute. One major computational difficulty in implementing procedures in [33] is to find, so called, a continuation region. A continuation region for a system is defined by upper and lower bounds such that the procedure will continue sampling from the system until a certain posterior parameter escapes the region. To find such a continuation region, one needs to use a backward algorithm to evaluate the value of information at each state of the posterior parameters. Since the state space of the posterior parameters is usually infinite (e.g., the posterior mean of normal sampling can be any real

number), the task of finding the continuation region cannot be done without discretization of the state space and approximation. Moreover, their procedures are practically useful only when variances are known or they are estimated only once. Otherwise, the continuation regions need to be recalculated whenever estimations of variances are updated, which is computationally very expensive. Finally, finding continuation regions is almost impossible under multiple constraints, because the dimension of the state space to be evaluated increases and the backward algorithm used in [33] only works for one dimension (i.e., a single constraint). Therefore, new Bayesian procedures that are more flexible and easier for implementation are desirable.

In the third part, we develop a novel feasibility determination procedure for discrete input space that is based on a Gaussian process (GP). It is natural to think that systems with similar configurations tend to show similar performance measures. A GP is commonly used by researchers on the input space as a surrogate of the objective function over an input space and try to capture such inherent relationship among systems with similar configurations. [38] provides a thorough overview of the use of Gaussian processes in the estimation of the objective function on the input space. The main idea is to assume that the objective function follows a GP and the decision makers choose priors for the mean and covariance matrix of the GP. Then a system or an alternative on the input space is evaluated and the posteriors of the mean and covariance matrix of the GP are updated. This idea is also found to be useful for optimization as in many existing works such as [23, 39, 40, 41, 24]. In Bayesian optimization, the role of the acquisition function is critical. Acquisition functions are designed to calculate how much potential improvement on the objective function exists at each input point and a fast convergence to a good or optimal solution can be achieved by choosing a point with the highest acquisition function value as the next point to evaluate. See [39] for popular acquisition functions in Bayesian optimization. We use a Gaussian process but with a goal of finding feasible regions in the presence of a single constraint. While the goal of Bayesian optimization is usually to find an input point with the largest

or smallest performance measure, our goal is to find input points whose performance measures are equal to a constraint threshold which is similar to finding zeros. Thus, typical acquisition functions do not work for our problem. In this thesis, we provide a new acquisition function appropriate for feasibility determination and incorporate it into a sequential procedure.

1.3 Spaitotemporal Scheduling

In the final part, we turn our focus to a spatiotemporal scheduling problem in manufacturing. It is well known that such a scheduling problem is challenging because (i) the number of decision variables is large due to a large number of blocks (usually 200–600) and a large number of possible process start time and placement decisions, and (ii) there are many constraints under consideration in order to have a feasible schedule. Major constraints related to this scheduling problem are as follows:

- Time window constraints: for each block, there is an associated time window and the block can only be assembled during that time window.
- Boundary constraints: since each bay has fixed location and dimensions, the placements of blocks are restricted to certain ranges.
- Production factor constraints: in each bay, many productivity factors, such as labors, numbers of tools required, capacities of equipments have their limits so that the assignments of blocks should not exceed those limits.
- Non-overlapping constraints: in each bay and on each date, the placements of blocks should not be overlapping with each other for safe production.
- Specialty constraints: some blocks require special skills or tools so that they are restricted to be assigned to certain bays.

- Additional constraints: other constraints may be needed to properly formulate the scheduling problem.

Many existing works in literature have been done on improving the block assembly process from different aspects. The study of assembly sequence planning (ASP) aims to find feasible and cost-effective sequences of blocks and operations for the assembly process. Many works use the knowledge-based approach, such as the case-based reasoning (CBR) method in [42, 43, 44, 45], and the rule-based reasoning (RBR) method in [46, 47]. The basic idea of the CBR method is to derive a solution to a new case by referring to the solution of a similar previous case and then adapting its solution appropriately to fit the new case such as [48], while the RBR method applies the expertise knowledge from past experience to design an ordering policy. Other methods used for solving the ASP problems utilize genetic algorithms as in [49, 50, 51], memetic algorithms in [52, 53], ant colony algorithms in [54, 55] and so on. From the literature, we see that the study of the ASP problems mainly focuses on the analysis of geometric constraints and precedence relations that must be satisfied in the assembly process while searching for the optimal sequence. Geometric constraints originate from the physical characteristics of the parts that are used to assemble the blocks.

Another main area of study related to the block assembly process is block spatial scheduling (BSS), which explores different placement strategies and algorithms to utilize space resource efficiently. Most of works on BSS have the objective as minimizing the time span of the assembly process. When a sequence of blocks for assembly is given, if a placement method can always fit as many blocks as possible in the bay, then the time span to finish assembling all the blocks can be minimized. Therefore, the study of BSS usually uses the time span to measure the effectiveness of a placement method. Many works on BSS mainly focus on developing efficient methods for block placements but use simple priority rules to determine the sequence of blocks. For example, [56] introduces the concept of largest contact area that aims to place more blocks in the bay and implement it in their BSS

algorithm. [57] proposes heuristic algorithms that place blocks using a hybrid of a bottom-left-fill method and a differential evolution algorithm. [58] develops a greedy scheduling algorithm that adopts a multi-step decision process for BSS and apply a different placement strategy on each step of the decision process. [59] proposes a heuristic block spatial scheduling model that adopts grid search and genetic algorithms. [60] modifies the bottom-left-fill method into a diagonal-fill method that considers both the bottom-left and top-right points for placements. [61] presents a branch and bound algorithm for spatial scheduling. There are some works that consider more sophisticated sequencing methods. [62] proposes a three-dimensional model for BSS by adding the time axis to the two-dimensional layout model, and use backtracking to make adjustments to the sequence. [63] decomposes the scheduling problem into two phases, in which the first phase assigns blocks to bays and determine the sequence and the second phase uses a tree search method for placements in each bay. [64] uses simulation to test different dispatching rules that control the block sequence along with their block placement system. [65] combines a genetic algorithm and a new spatial allocation policy based on, so called, the best contact area to search for an optimal block assembly schedule. There are other existing works that consider outsourcing. Using a constraint-based model, [66] develops a branch and bound algorithm to provide temporal schedules and bay assignments for the block assembly process with objectives to minimize the number of outsourced blocks and discrepancy of workloads across different bays and dates. Considering both the block assembly process and the outfitting process (i.e., installations of equipments, piping, ducting and cabling in the assembled blocks), [67] uses a genetic algorithm to find temporal schedules that minimize the number of outsourced blocks and variations of resource requirements.

Although there exists a considerable amount of works in the literature, none of the existing methods are directly applicable to the scheduling problem we consider in this thesis. We want to perform both temporal and spacial scheduling on the assembly process of grand blocks with objectives to minimize the number of outsourced blocks and to bal-

ance the workload distribution. However, the existing methods consider either different time and space constraints or different objectives than ours. As a result, a company like Hyundai Heavy Industries (HHI) currently determines block assembly schedules manually, which can easily takes up to two weeks. Even if a manual schedule can be found, it is hard to verify whether the resulting schedule is feasible through human efforts, due to the large scale of the problem. Furthermore, manually updating the schedule according to the current progress can be inconvenient, as more human efforts are needed to resolve the scheduling problem. Consequently, it is desirable to have an automatic scheduling tool that minimizes the number of outsourced blocks and provides time information on when to assemble each block and placement information on where to place each block. More specifically, for each block, the time information should tell the start date and finish date to assemble the block, and the placement information should include its assigned bay, the coordinate for placement on the bay and the rotation degree. In addition, it would be desirable that the scheduling approach can take information of current progress such as blocks that are already under assembly and outsourced blocks as prior information and find an updated schedule in a relatively short amount of time.

In this thesis, we propose a heuristic two-phase approach to solve the spatiotemporal scheduling problem and test our procedure using datasets provided by HHI.

The remaining of this thesis is organized as follows: In Chapter 2, we present a new form of reward function to be used in Bayesian feasibility determination procedures, and compare its performance against other reward functions and also types of procedures. In Chapter 3, we present new sequential Bayesian procedures that are easy for computation and have competitive performances. In Chapter 4, we present a new Bayesian procedure based on Gaussian process. In Chapter 5, we present a two-phase approach for spatiotemporal scheduling, with a case study in shipbuilding.

CHAPTER 2
A NEW REWARD FUNCTION FOR BAYESIAN FEASIBILITY DETERMINATION PROCEDURES

In Bayesian feasibility determination, a typical reward function is either the 0-1 or linear reward function. In this chapter, we propose a new type of reward function for Bayesian feasibility determination. Our proposed reward function emphasizes the importance of barely feasible/infeasible systems whose mean performance measures are close to the threshold. Here, we define a system as barely feasible/infeasible if its mean performance measure is close to a threshold value of the given constraint, and clearly feasible/infeasible if the mean performance measure is far from the threshold. There are two main reasons why the barely feasible/infeasible systems are more important. First, the overall accuracy on solving a feasibility determination problem is heavily affected by those difficult systems. Second, if the decision maker wants to further find the best feasible system, it is likely that one of the barely feasible/infeasible systems is the best feasible. We derive a feasibility determination procedure with the new reward function in a Bayesian framework.

In Section 2.1, we provide our notation and assumptions, and define the feasibility determination problem. Then, in Section 2.2, we present our new reward function, provide the corresponding Bayesian optimal policy, and specify the actual procedure to be used in practice; in Section 2.3, we present results from illustrative experiments to show the advantage of our proposed procedure in solving the feasibility determination problem, followed by a conclusion in Section 2.4.

2.1 Background

In this section, we introduce notation and necessary assumptions on simulation processes and define the feasibility determination problem.

2.1.1 Assumptions

For a simulation process, let $\mu_i = \mathbb{E}[Y_{ij}] \in \mathbb{R}$ and $\gamma_i = 1/\text{Var}[Y_{ij}] \in (0, \infty)$, where Y_{ij} represents the jth simulation observation from system design i, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots$. We make the following assumptions on the simulation process:

Assumption 1. *For any two systems $i, i' \in \{1, 2, \ldots, k\}$ such that $i \neq i'$ and $j = 1, 2, \ldots$ and $j' = 1, 2, \ldots$, Y_{ij} and $Y_{i'j'}$ are independent.*

Assumption 1 means that common random numbers are not used in the simulation process and observations from different systems are mutually independent.

Assumption 2. *For each system $i = 1, 2, \ldots k$, $Y_{ij} \overset{iid}{\sim} \mathcal{N}(\mu_i, 1/\gamma_i)$ for $j = 1, 2, \ldots$.*

Throughout the thesis, we denote $\mathcal{N}(p, q)$ as a normal distribution with mean p and variance q. Assumption 2 is plausible if Y_{i1},$Y_{i2}, \ldots$ are within-replication averages across independent replications of system i, or if they are batch means with a large batch size from a single replication of a steady-state simulation after accounting for initialization effects. For more information, see [3].

The sampling precisions γ_i's are assumed to be known in this chapter (as in many other Bayesian R&S works). However, we consider μ_i's as the unknown mean performance measures of interest, for $i = 1, 2, \ldots, k$. Using the Bayesian approach, we place a prior distribution on each μ_i. We suppose that these prior distributions come from the same distribution family ζ with parameter space Ω. To facilitate computation, we adopt independent conjugate priors. Specifically, we have the following assumption:

Assumption 3. *For $i = 1, 2, \ldots, k$, μ_i's are mutually independent and $\mu_i \sim \mathcal{N}(\eta_i, 1/\lambda_i)$, where $\eta_i = \mathbb{E}[\mu_i] \in \mathbb{R}$ and $\lambda_i = 1/\text{Var}[\mu_i] \in (0, \infty)$.*

The assumption of known sampling precision γ_i is rarely true in real practice. The frequentist's approach, such as IZ approach, tends to deal with unknown γ_i's directly. However, the OCBA and Bayesian approaches often work on versions for known precisions.

They then address the unknown variances by running a first-stage experiment that simulates a small number n_0 of replications and estimates γ_i by using its maximum likelihood estimator. In this chapter, we consider known sampling precisions only.

2.1.2 Problem Formulation

In general, the goal of a feasibility determination problem is to find a set of systems among a finite number of simulated systems. Without loss of generality, we define that a system is feasible if and only if its mean performance measure of interest is less than or equal to the corresponding threshold. For simplicity, we consider situations where there is only one constraint with a threshold d. We consider there are k available systems and there is a fixed unit cost c associated with simulating an observation from any system. Under Assumptions 2 and 1, each $Y_{ij} \overset{iid}{\sim} \mathcal{N}(\mu_i, 1/\gamma_i)$, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots$. Therefore, a system i is feasible if and only if $\mu_i \leq d$ for a threshold d. We define $\mathbb{F} = \{i : \mu_i < d\}$, which is the true set of feasible systems.

The stage is indexed by $n = 0, 1, 2, \ldots$. At each stage n, we choose exactly one system $i_n \in \{1, 2, \ldots, k\}$ to sample, and let $S_{n,i}$ be the parameters of the posterior distribution for $i = 1, 2, \ldots, k$. By convention, we denote $S_{0,i}$ as the parameters of prior distribution for μ_i. Under Assumption 3, since we choose priors that are conjugate to the sampling distributions, the sampling process results in a sequence of posterior distributions, each of which resides in the same distribution family ζ parameterized by the same space Ω. Therefore, we have that $S_{n,i} = (\eta_{n,i}, \lambda_{n,i}) \in \Omega$, where $\Omega = \mathbb{R} \times (0, \infty)$, for all $n = 0, 1, 2, \ldots$ and $i = 1, 2, \ldots, k$.

The reward function r is chosen by the decision maker to represent importance when a correct decision on feasibility is made. Specifically, the reward function r is defined as a two-piece function:

$$r(F; \boldsymbol{\mu}, d) = \sum_{i \in F} r_0(\mu_i, d) + \sum_{i \notin F} r_1(\mu_i, d),$$

where r_0 and r_1 are known real-valued functions, F is any subset of $\{1, 2, \ldots, k\}$, and $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_k\}$.

At each stage n, the set $F_n \subset \{1, \ldots, k\}$ is chosen to maximize the expected reward function, given the information of n observations. Specifically, for all $n \geq 0$,

$$\begin{aligned} F_n &= \underset{F \subset \{1,2,\ldots,k\}}{\arg\max}\ \mathbb{E}_n \left[r(F; \boldsymbol{\mu}, d) \right] \\ &= \underset{F \subset \{1,2,\ldots,k\}}{\arg\max} \left\{ \sum_{i \in F} \mathbb{E}_n \left[r_0(\mu_i, d) \right] + \sum_{i \notin F} \mathbb{E}_n \left[r_1(\mu_i, d) \right] \right\}. \end{aligned}$$

Eventually, the estimate of $\mathbb{F}$ returned by the procedure is F_τ, where τ is the stopping time determined by the policy. Our goal is to find a policy π^* that maximizes the expected total reward. Specifically, we want to solve the problem

$$\sup_{\pi} \mathbb{E}^{\pi} \left[r(F_\tau; \boldsymbol{\mu}, d) - \tau c \right]. \tag{2.1}$$

For reference, Table 2.1 summarizes notation used throughout this chapter.

2.2 An Alternative Choice of Reward Function

As discussed in section 2.1.2, the decision maker needs to specify a reward function in order to find an optimal policy. Two common choices are 0-1 and linear reward functions, which take the form as follows:

- the 0-1 reward function: $r_0(\mu, d) = \mathrm{I}(\mu \leq \mathrm{d})$; $r_1(\mu, d) = \mathrm{I}(\mu > \mathrm{d})$, where $\mathrm{I}(\cdot)$ is the indicator function;
- the linear reward function: $r_0(\mu, d) = d - \mu$; $r_1(\mu, d) = \mu - d$.

In this section, we purpose a new reward function, so called *normal reward function*. We provide its corresponding Bayesian optimal policy, based on work from [33]. At the

Table 2.1: Summary of notation.

notation	meaning
k	total number of available systems
d	control requirement for each system i, $i = 1, \ldots, k$
c	cost per simulation for each system i, $i = 1, \ldots, k$
n	stage counter, $n = 0, 1, 2, \ldots$
μ_i	mean performance for system i, $i = 1, \ldots, k$
$\boldsymbol{\mu}$	vector of means $(\mu_1, \mu_2, \ldots, \mu_k)$
γ_i	sampling precision for system i, $i = 1, \ldots, k$
ζ	distribution family of prior and posterior distributions
Ω	parameter space of prior and posterior distributions
$\eta_{0,i}$	mean of prior distribution on μ_i, $i = 1, \ldots, k$
$\eta_{n,i}$	mean of posterior distribution on μ_i at stage $n = 1, 2, \ldots$, $i = 1, \ldots, k$
$\lambda_{0,i}$	precision of prior distribution on μ_i, $i = 1, \ldots, k$
$\lambda_{n,i}$	precision of posterior distribution on μ_i at stage $n = 1, 2, \ldots$, $i = 1, \ldots, k$
$S_{n,i}$	state of parameters of distribution on μ_i; $S_{n,i} = (\mu_{n,i}, \lambda_{n,i})$
$\boldsymbol{S_n}$	vector of states $(S_{n,1}, S_{n,2}, \ldots, S_{n,k})$
π	policy that governs the rules of sampling and termination
τ	stopping time determined by the policy
$\mathbb{F}$	true set of feasible systems, $\mathbb{F} \subset \{1, 2, \ldots, k\}$
F_n	estimate of $\mathbb{F}$ at stage $n = 0, 1, 2, \ldots$
F_τ	final estimate of $\mathbb{F}$ returned by the procedure

end, we present an example procedure built on the Bayesian optimal policy to be used in practice.

2.2.1 Normal Reward Function

As discussed in Chapter 1, barely feasible/infeasible systems are often more important in the feasibility determination problem. However, neither 0-1 nor linear can capture such importance of barely feasible/infeasible systems. Therefore, we purpose the normal reward function $r(F; \boldsymbol{\mu}, d) = \sum_{i \in F} r_0(\mu_i, d) + \sum_{i \notin F} r_1(\mu_i, d)$, where r_0 and r_1 are as follows:

$$r_0(\mu_i, d) = \begin{cases} a \cdot \exp\left\{-\frac{1}{2}(d - \mu_i)^2 \cdot b\right\}, & \text{if } \mu_i \leq d; \\ 0, & \text{otherwise;} \end{cases}$$

$$r_1(\mu_i, d) = \begin{cases} 0, & \text{if } \mu_i \leq d; \\ a \cdot \exp\left\{-\frac{1}{2}(d-\mu_i)^2 \cdot b\right\}, & \text{otherwise.} \end{cases}$$

The normal reward function assigns reward values that follow the shape of a normal distribution, with its maximum at the threshold. As a result, barely feasible/infeasible systems tend to have larger rewards than clearly feasible/infeasible ones. In addition, there are two parameters that the decision maker must choose before implementation. Generally speaking, the parameter a determines the maximum magnitude of the reward, and b determines the spread-out of the reward.

2.2.2 Conditions on Reward Functions

Although the problem formulation can apply to general reward functions, our method to find the Bayesian optimal policy requires additional conditions on the reward functions. Before proceeding to the next section, we want to state these conditions.

First of all, we define the functions that are useful in deriving the optimal policy in the next section. For any $s \in \Omega$, define

$$\begin{aligned} h_{0i}(s) &= \mathbb{E}\left[r_0(\mu, d)|\mu \sim \zeta(s)\right]; \\ h_{1i}(s) &= \mathbb{E}\left[r_1(\mu, d)|\mu \sim \zeta(s)\right]; \\ h_i(s) &= \max\left\{h_{0i}(s), h_{1i}(s)\right\}; \\ R_i(s) &= \mathbb{E}[h_i(S_{1,i})|S_{0,i} = s, i_1 = i] - h_i(s) - c; \\ V_i(s) &= \sup_{\tau_i} \mathbb{E}^{\tau_i}\left[\sum_{n=1}^{\tau_i} R_i(S_{n-1,i})|S_{0,i} = s, i_1 = \ldots = i_{\tau_i} = i\right]. \end{aligned}$$

Note that we use τ_i to represent the stopping time for system i, but also the policy in which only system i can be sampled and it is sampled for τ_i times.

The function $h_i(s)$ can be interpreted as the expected reward to be obtained with respect

to a distribution of the unknown performance measure. The function $R_i(s)$ represents the value of information (VOI) of one additional sampling, which can also be interpreted as the expected increment of expected reward given one more sample.

The required conditions are then as follows:

Condition 1. *For each system i, there exists a deterministic nonnegative function $H_i(s)$ on Ω such that for any $s \in \Omega$,*

$$\mathbb{E}\left[h_i(S_{n,i})|S_{0,i} = s, i_1 =, \ldots, i_n = i\right] - h_i(s) \leq H_i(s).$$

Condition 2. *For each system i, there exists a deterministic nonnegative function $\tilde{H}_i(s)$ on Ω such that for any $s \in \Omega$,*

$$\mathbb{E}\left[h_i(S_{1,i})|S_{0,i} = s, i_1 = i\right] - h_i(s) \leq \tilde{H}_i(s),$$

$$\lim_{n\to\infty}\left[\sup_{s\in PS(i,;n)} \tilde{H}_i(s)\right] = 0,$$

where

$$PS(i;n) = \{s \in \Omega : \exists s' \in \Omega \text{ s.t. } \Pr\{S_{n,i} = s|S_{0,i} = s', i_1 = i_2 = \ldots = i_n = i\} > 0\}.$$

Condition 3. *For any system i and precision λ, there exists an interval $\left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right]$ such that $\eta \notin \left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right] \Rightarrow V_i(\eta, \lambda) = 0$.*

Proofs that the normal reward function satisfies the above conditions are in the Appendix A.1.

2.2.3 Bayesian Optimal Sampling Policy

The problem (2.1) can be solved using dynamic programming techniques. First, we define a value function $V : \Omega^k \to \mathbb{R}$. For each $\boldsymbol{s} \in \Omega^k$, $V(\boldsymbol{s})$ is the optimal expected total reward

attainable when the initial state is $\boldsymbol{s}$. Specifically,

$$V(\boldsymbol{s}) = \sup_{\pi} \mathbb{E}^{\pi} \left[r(F_{\tau}; \boldsymbol{\mu}, d) - \tau c | \boldsymbol{S}_0 = \boldsymbol{s} \right]. \tag{2.2}$$

[33] proves that (2.2) is equivalent to

$$V(\boldsymbol{s}) = \sup_{\pi} \mathbb{E}^{\pi} \left[\sum_{n=1}^{\tau} R_{i_n}(S_{n-1,i_n}) | \boldsymbol{S}_0 = \boldsymbol{s} \right]. \tag{2.3}$$

Instead of solving (2.3) directly, consider the subproblem where only system i can be sampled. Specifically, the subproblem is

$$V_i(s) = \sup_{\tau_i} \mathbb{E}^{\tau_i} \left[\sum_{n=1}^{\tau_i} R_i(S_{n-1,i}) | S_{0,i} = s, i_i = \ldots = i_{\tau_i} = i \right]. \tag{2.4}$$

Results from the dynamic programming literature (see [68]) show that $V_i(s)$ satisfies Bellman's recursion,

$$V_i(s) = \max \left[0, L_i(s, V_i) \right], \tag{2.5}$$

where $L_i(s, V_i) = R_i(s) + \mathbb{E}\left[V_i(S_{1,i}) | S_{0,i} = s, i_1 = i \right]$.

The problem (2.5) is a standard optimal stopping problem that can be solved by specifying the so-called continuation set $\mathbb{C}_i$ (see [69]), defined as $\mathbb{C}_i = \{s \in \Omega : V_i(s) > 0\}$. Then, an optimal solution to (2.4) is the stopping time τ_i^* given by $\tau_i^* = \inf \{n \geq 0, S_{n,i} \notin \mathbb{C}_i\}$. In general, τ_i^* can go to ∞. However, under Condition 2, we can provide a deterministic upper bound on τ_i^*, denoted as N_i, as follows:

$$N_i = \min \left\{ n : \left[\sup_{s \in PS(i,;n')} \tilde{H}_i(s) \right] \leq c, \forall n' \geq n \right\}. \tag{2.6}$$

Given that all systems are mutually independent, it is straightforward that $V(\boldsymbol{s}) = \sum_{i=1}^{k} V_i(s_i)$, where $\boldsymbol{s} = (s_1, \ldots, s_k)$. Furthermore, we have the following theorem:

Theorem 1. *Any policy π^* with sampling decisions $(i_1^*, i_2^*, \ldots, i_n^*)$ and stopping time τ^**

satisfying the following is optimal:

$$i^*_{n+1} \in \{i, S_{n,i} \in \mathbb{C}_i\}, \forall i \geq 0;$$
$$\tau^* = \inf\{n \geq 0 : S_{n,i} \notin \mathbb{C}_i, \forall i\}.$$

Proof of Theorem 1 can be found in [33]. Since the value of information from each stage only depends on the system being sampled and the states of other systems remain unchanged, the order of the sequence of sampling decisions does not affect the total value of information. Therefore, we can solve each subproblem (2.4) separately, and the final optimal policy π^* is found based on Theorem 1 with $\tau^* = \sum_{i=1}^{k} \tau_i^*$.

We can then focus on solving (2.4). Under Assumptions 2 and 1, [33] provides a probabilistically equivalent form of the update of the posterior parameters:

$$\mathbb{E}\left[V_i(S_{1,i})|S_{0,i} = s, i_1 = i\right] = \mathbb{E}\left[V_i(\eta + \tilde{\sigma}_i(\lambda)Z, \lambda + \gamma_i)\right],$$

where $s = (\eta, \lambda)$ for $\eta \in \mathbb{R}$ and $\lambda \in (0, \infty)$, $\tilde{\sigma}_i(\lambda) = \sqrt{\frac{\gamma_i}{\lambda(\lambda+\gamma_i)}}$ and Z is a standard normal random variable. Therefore, (2.5) becomes

$$V_i(\eta, \lambda) = \max\left[0, L_i(\eta, \lambda, V_i)\right], \tag{2.7}$$

where $L_i(\eta, \lambda, V_i) = R_i(\eta, \lambda) + \mathbb{E}\left[V_i(\eta + \tilde{\sigma}_i(\lambda)Z, \lambda + \gamma_i)\right]$.

To calculate $V_i(\eta, \lambda)$ for all possible $(\eta, \lambda) \in \Omega$, the main idea is to use a backward algorithm. First, we assume that after a large number of stages N_i, $V_i(\eta, \lambda_{0,i} + n\gamma_i) = 0$, for all $n > N_i$ and all $\eta \in \mathbb{R}$. The number N_i can be found by using (2.6). However, for simplicity, we set $N_i = 1000$ for the experiments. Starting from $\lambda = \lambda_{0,i} + N_i\gamma_i$, we compute $\left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right]$ as the boundary of η such that $V_i(\eta, \lambda) = 0$, if $\eta \notin \left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right]$. Under Condition 3, we know such $\left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right]$ exists. Then, we discretize the range $\left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right]$ into points $\{\eta_i(\lambda)^j\}$ with an interval of δ between them (in our experiments,

we set $\delta = 0.01$). Using (2.7) and an approximation:

$$\mathbb{E}\left[V_i\left(\eta + \tilde{\sigma}_i(\lambda)Z, \lambda + \gamma_i\right)\right] \approx \sum_j V_i\left(\eta_i^j(\lambda+\gamma_i), \lambda+\gamma_i\right) \cdot \left[\Phi\left(\frac{\eta_i^j(\lambda+\gamma_i) + \delta/2 - \eta}{\tilde{\sigma}(\lambda)}\right) - \Phi\left(\frac{\eta_i^j(\lambda+\gamma_i) - \delta/2 - \eta}{\tilde{\sigma}(\lambda)}\right)\right],$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard normal random variable, each $V_i(\eta_i(\lambda)^j, \lambda)$ can be computed recursively for $\lambda \in \{\lambda_{0,i} + n\gamma_i : 0 \le n \le N\}$.

Finally, for any arbitrary $(\eta, \lambda) \in \mathbb{R} \times \{\lambda_{0,i} + n\gamma_i : 0 \le n \le N\}$, we set

$$V_i(\eta, \lambda) = \begin{cases} 0, & \text{if } \eta \notin \left[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)\right]; \\ V_i\left(\eta_i^j(\lambda), \lambda\right), & \text{otherwise, where } j = \arg\min\left\{|\eta - \eta_i^j(\lambda)|\right\}. \end{cases}$$

As a result, we find $\mathbb{C}_i = \left\{[\overline{\eta_i}(\lambda), \underline{\eta_i}(\lambda)] : \lambda = \lambda_{0,i} + n\gamma_i, 0 \le n \le N\right\}$.

The remaining work to complete the policy is to specify the $h_i(\eta, \lambda)$ and $R_i(\eta, \lambda)$ functions, for each $i = 1, 2, \ldots, k$. First, we start with $h_{0i}(\eta, \lambda)$ and $h_{1i}(\eta, \lambda)$:

$$\begin{aligned} h_{0i}(\eta, \lambda) &= \mathbb{E}\left[r_0(\mu, d) | \mu \sim \mathcal{N}(\eta, 1/\lambda)\right] \\ &= \int_{-\infty}^{d} a \cdot \exp\left\{-\frac{1}{2}(d-\mu)^2 b\right\} \cdot \frac{\sqrt{\lambda}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\mu-\eta)^2\lambda\right\} d\mu \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}} \int_{-\infty}^{d} \frac{\sqrt{b}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(d-\mu)^2 b\right\} \cdot \frac{\sqrt{\lambda}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\mu-\eta)^2\lambda\right\} d\mu \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta, b, \lambda) \int_{-\infty}^{d} \frac{1}{\sqrt{2\pi}} \sqrt{b+\lambda} \exp\left\{-\frac{1}{2}(\mu - \frac{db+\eta\lambda}{b+\lambda})^2 (b+\lambda)\right\} d\mu \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta, b, \lambda) \cdot \Phi\left((d - \frac{db+\eta\lambda}{b+\lambda})\sqrt{b+\lambda}\right), \end{aligned}$$

where

$$G(\eta_f, \eta_g, \lambda_f, \lambda_g) = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{\lambda_f \lambda_g}{\lambda_f + \lambda_g}} \exp\left\{-\frac{1}{2}(\eta_f - \eta_g)^2 \frac{\lambda_f \lambda_g}{\lambda_f + \lambda_g}\right\}.$$

Using similar argument,

$$\begin{aligned} h_{1i}(\eta,\lambda) &= \mathbb{E}\left[r_1(\mu,d)|\mu\sim\mathcal{N}(\eta,1/\lambda)\right] \\ &= \int_d^\infty a\cdot\exp\left\{-\frac{1}{2}(d-\mu)^2 b\right\}\cdot\frac{\sqrt{\lambda}}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}(\mu-\eta)^2\lambda\right\}d\mu \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}}G(d,\eta,b,\lambda)\cdot\left[1-\Phi\left((d-\frac{db+\eta\lambda}{b+\lambda})\sqrt{b+\lambda}\right)\right]. \end{aligned}$$

Finally,

$$h_i(\eta,\lambda) = \max\left\{h_{0i}(\eta,\lambda), h_{1i}(\eta,\lambda)\right\}.$$

The computation of $R_i(\eta,\lambda)$ is more complicated. Using the equivalent form of the update of posterior parameters, $R_i(\eta,\lambda)$ can be expressed as:

$$R_i(\eta,\lambda) = \mathbb{E}\left[h_i(\eta+\tilde{\sigma}_i(\lambda)\cdot Z, \lambda+\gamma_i)\right] - h_i(\eta,\lambda) - c.$$

Since h_i is well defined and c is given, we only need to find $\mathbb{E}[h_i(\eta+\tilde{\sigma}_i(\lambda)\cdot Z,\lambda+\gamma_i)]$. The result is as follows, and details of calculation are in Appendix A.2.

$$\begin{aligned} &\mathbb{E}\left[h_i(\eta+\tilde{\sigma}_i(\lambda)\cdot Z,\lambda+\gamma_i)\right] = \\ &\frac{a\sqrt{2\pi}}{\sqrt{b}}\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}}G\left(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)},0,\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)},1\right)\cdot\mathbf{Pr}\left\{Z_1\le\frac{d-\eta}{\tilde{\sigma}_i(\lambda)},Z_3\le 0\right\} \\ &+\frac{a\sqrt{2\pi}}{\sqrt{b}}\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}}G\left(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)},0,\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)},1\right)\cdot\mathbf{Pr}\left\{Z_1\ge\frac{d-\eta}{\tilde{\sigma}_i(\lambda)},Z_3\ge 0\right\}, \end{aligned}$$

where $\begin{bmatrix} Z_1 \\ Z_3 \end{bmatrix}$ follows a multivariate normal distribution $MVN(\tilde{\mathbf{m}},\tilde{\mathbf{\Sigma}})$ with mean $\tilde{\mathbf{m}}$ and covariance matrix $\tilde{\mathbf{\Sigma}}$, and

$$\tilde{\mathbf{m}} = \begin{bmatrix} m_1 \\ -A+Bm_1 \end{bmatrix},$$

$$\tilde{\mathbf{\Sigma}} = \begin{bmatrix} 1/b_1 & B/b_1 \\ B/b_1 & 1 + B^2/b_1 \end{bmatrix},$$

with $m_1 = \frac{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)} b\gamma_i}{b\gamma_i + \lambda(b+\lambda+\gamma_i)}$, $b_1 = \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)} + 1$, $A = -\frac{(d-\eta)(\lambda+\gamma_i)}{\sqrt{b+\lambda+\gamma_i}}$ and $B = \frac{\tilde{\sigma}_i(\lambda)(\lambda+\gamma_i)}{\sqrt{b+\lambda+\gamma_i}}$. Therefore, $\mathbb{E}\left[h_i(\eta + \tilde{\sigma}_i(\lambda) \cdot Z, \lambda + \gamma_i)\right]$ can be calculated using the cumulative distribution function of a multivariate normal distribution.

2.2.4 The $\mathcal{XF}$ Procedure

In this section, we present an example procedure, denoted as $\mathcal{XF}$, where the normal reward function can be implemented. The $\mathcal{XF}$ procedure is based on work from [33]. As discussed in Section 2.2.3, the problem (2.1) can be decomposed into subproblems, each of which has the form as problem (2.4). For each system $i = 1, 2, \ldots, k$, the procedure first computes each $\mathbb{C}_i$. Then, it keeps sampling from system i and updating $S_{n_i,i}$, until $S_{n_i,i}$ is out of the set $\mathbb{C}_i$. The way to update $S_{n_i,i}$ follows the Bayesian rules. Specifically, the $\mathcal{XF}$ procedure is stated in Algorithm 1.

Algorithm 1 $\mathcal{XF}$ procedure

1: **Setup**: Let $F = \emptyset$. Specify number of systems k, threshold d and unit cost c. Start with system $i = 1$.
2: **Initialization**: Specify prior distribution $\mathcal{N}(\eta_{0,i}, 1/\lambda_{0,i})$ for the mean performance sampling precision γ_i and cost per unit c. Compute continuation region $\mathbb{C}_i$. Set $n_i = 0$.
3: **Update**: Let $n_i = n_i + 1$. Simulate one observation y_i from system i. Compute

$$\eta_{n_i,i} = \frac{\lambda_{n_i-1,i}\eta_{n_i-1,i} + \gamma_i y_i}{\lambda_{n_i-1,i} + \gamma_i},$$

$$\lambda_{n_i,i} = \lambda_{n_i-1,i} + \gamma_i.$$

4: **Stopping Rule**: If $(\eta_{n_i,i}, \lambda_{n_i,i}) \notin \mathbb{C}_i$, then stop sampling from system i and go to **Feasibility Check**. Otherwise, go back to **Update**.
5: **Feasibility Check**: If $\eta_{n_i,i} < d$, then add i in F.
6: **Termination Rule**: Set $i = i + 1$. If $i \leq k$, go back to **Initialization**. Otherwise, return F as the set of feasible systems.

To finish up the procedure, we want to discuss how to specify the parameters in the

normal reward function. For the parameter b, we set $b = \lambda_{0,i}$. Since $\lambda_{0,i}$ represents our initial knowledge about the spread-out of μ_i, setting $b = \lambda_{0,i}$ means the distribution of reward has a similar shape as that of μ_i. To decide the value of a, we equalize the initial expected rewards from 0-1 and normal reward functions and solve for a, because we want to compare the two reward functions and assume that the decision maker expects the same amount of reward from both reward functions. Based on our settings, this is to solve

$$\frac{a\sqrt{2\pi}}{\sqrt{\lambda_{0,i}}} G(d, \eta_{0,i}, \lambda_{0,i}, \lambda_{0,i}) \cdot \frac{1}{2} = \frac{1}{2}$$

and get $a = \sqrt{2}$.

2.3 Numerical Experiments

In this section, we provide illustrative experiments to demonstrate the advantages of using the Bayesian approach with the normal reward function. In our experiments, we want to make two types of comparisons. First, we make a comparison among three different reward functions: the normal, 0-1 and linear reward functions. Here, we denote the three Bayesian procedures using normal, 0-1 and linear reward functions as $\mathcal{XF}$ normal, $\mathcal{XF}$ 0-1 and $\mathcal{XF}$ linear, respectively. Second, we make a comparison among procedures from different approaches:

- The $\mathcal{XF}$ normal procedure. As discussed above, it is based on the Bayesian approach and utilizes the normal reward function.
- The $\mathcal{BK}$ procedure from [28]. This procedure falls in a category of IZ procedures. Instead of an IZ parameter, it uses an error tolerance which is similar to the concept of the IZ parameter in the finding-the-best problem.
- The $\mathcal{GC}$ procedure from [32]. This procedure uses the OCBA framework and is based on the large deviation principle.

More details about the $\mathcal{BK}$ and $\mathcal{GC}$ procedures can be found in Appendix A.3 and A.4.

To measure the performance of a procedure, we need to introduce some metrics. We define CD_i as the event of making a correct decision on feasibility of system i, and $\text{CD} \equiv \cap_{i=1}^{k} \text{CD}_i$ as the event of making correct decisions on feasibilities of all available systems. Furthermore, we define $\text{PCD}_i \equiv \Pr\{\text{CD}_i\}$ and $\text{PCD} \equiv \Pr\{\text{CD}\}$. In our experiments, we estimate each PCD_i and PCD empirically based on 10000 replications. We take PCD as our main metric in measuring the performance of a procedure because it represents the overall accuracy in solving the feasibility determination problem, while we also keep track of each PCD_i to gain insights on its system-wise performance. At last, we record the average number of observations spent on each system per replication, and find the average total number of observations per replication. In general, a procedure has good performance if it returns a high PCD, while its average total number of observations per replication is small.

2.3.1 Experimental Settings

We consider $k = 50$ systems. Without loss of generality, we set the threshold $d = 0$ for all systems. For simplicity, the unit cost of simulation is $c = 0.001$ for all systems. Under Assumptions 2 and 1, each observation Y_{ij} is independently simulated from $\mathcal{N}(\mu_i, 1/\gamma_i)$. For each μ_i, we place a conjugate prior distribution $\mu_i \sim \mathcal{N}(\eta_{0,i}, 1/\lambda_{0,i})$ with $\eta_{0,i} = 0$ and $\lambda_{0,i} = 0.01$. The true mean performances μ_i of systems are ranged from -4.5 to 4.5 as in Table 2.2. The set $\mathbb{F} = \{1, 2, \ldots, 25\}$ is the true set of feasible systems.

We consider three configurations for the systems' true precisions γ_i, $i = 1, 2, \ldots, 50$: constant precisions (CP), decreasing precisions (DP) and increasing precisions (IP). In CP, we set $\gamma_i = 1$ for all systems. As the true mean performances move away from the standard, the systems' true precisions decrease in DP, while they increase in IP. In particular, $\gamma_i = 1/[1 + (|i - 25.5| - 0.5) \cdot 0.1]^2$ for $i = 1, 2, \ldots$ in DP; $\gamma_i = [1 + (|i - 25.5| - 0.5) \cdot 0.1]^2$ for $i = 1, 2, \ldots$ in IP.

For the $\mathcal{BK}$ procedure, we use a simple grid search to explore different values for the

Table 2.2: Systems' true mean performances.

i	μ_i	i	μ_i	i	μ_i	i	μ_i	i	μ_i
1	-4.5	11	-1.6	21	-0.3	31	0.4	41	1.8
2	-4	12	-1.4	22	-0.25	32	0.5	42	2
3	-3.5	13	-1.2	23	-0.2	33	0.6	43	2.2
4	-3	14	-1	24	-0.15	34	0.7	44	2.4
5	-2.8	15	-0.9	25	-0.1	35	0.8	45	2.6
6	-2.6	16	-0.8	26	0.1	36	0.9	46	2.8
7	-2.4	17	-0.7	27	0.15	37	1	47	3
8	-2.2	18	-0.6	28	0.2	38	1.2	48	3.5
9	-2	19	-0.5	29	0.25	39	1.4	49	4
10	-1.8	20	-0.4	30	0.3	40	1.6	50	4.5

confidence level $1-\alpha$ and error tolerance ϵ to find appropriate settings such that the procedure produces approximately the same average total number of observations as the $\mathcal{XF}$ normal procedure. The reason for doing so is that we can compare the two procedures by comparing their PCD, while keeping the total cost roughly the same. Based on the grid search, the settings are as presented in Table 2.3

Table 2.3: Choices of nominal confidence level $1-\alpha$ and error tolerance ϵ for $\mathcal{BK}$ in the CP, DP and IP configurations.

	$1-\alpha$	ϵ
CP	0.72	0.20
DP	0.60	0.35
IP	0.92	0.18

For the $\mathcal{GC}$ procedure, we set the total budget equal to the average total number of observations per replication of the $\mathcal{XF}$ normal procedure in each configuration. Again, in this way, we can compare the two procedures by their PCD, given the total cost is exactly the same. In addition, we set its initial sample size $n_0 = 2$ for each system, and the incremental budget at each stage $\Delta_0 = 5$.

2.3.2 Results

For the comparison among different reward functions, Table 2.4 shows that $\mathcal{XF}$ normal has the highest PCD, 0-1 the second, and linear the lowest, in all three configurations. From

Figure 2.1, we see that the improvement on PCD by using the normal reward function is due to more correct decisions on feasibilities of the barely feasible/infeasible systems (i.e., systems 20 to 30). The relatively poor performance from using the linear reward function is expected. As discussed in Chapter 1, although the linear reward function is ideal for modeling opportunity cost, it is not suitable for feasibility determination. In addition, we note that to achieve a higher PCD by using the normal reward function, a larger average total number of observations per replication is spent, based on Table 2.5. However, Figure 2.1 shows that the normal reward spends substantially more observations only on the barely feasible/infeasible systems. This result is desirable, as the procedure treats the barely feasible/infeasible systems as more important than the clearly feasible/infeasible ones and therefore, and spend more efforts in making decisions on their feasibilities to achieve a higher PCD.

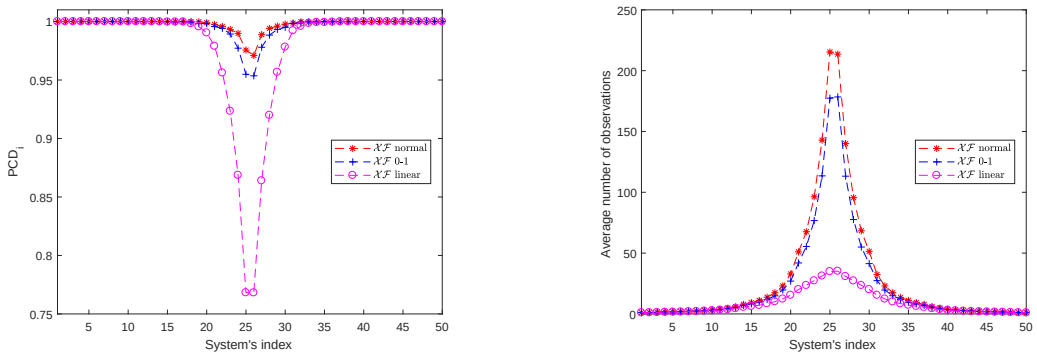
Table 2.4: Summary of PCD by using $\mathcal{XF}$ normal, $\mathcal{XF}$ 0-1 and $\mathcal{XF}$ linear procedures, under CP, DP and IP configurations.

	$\mathcal{XF}$ normal	$\mathcal{XF}$ 0-1	$\mathcal{XF}$ linear
CP	0.9004	0.8256	0.3256
DP	0.8412	0.7152	0.2425
IP	0.9258	0.8727	0.3894

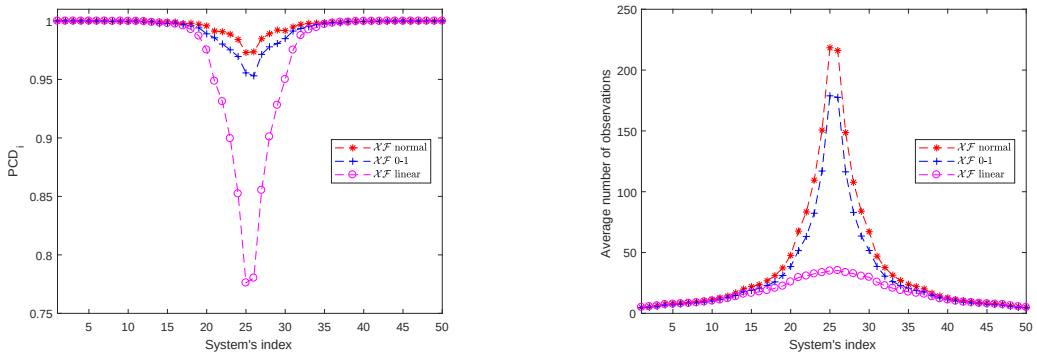
Table 2.5: Summary of average total number of observations taken by the $\mathcal{XF}$ normal, $\mathcal{XF}$ 0-1 and $\mathcal{XF}$ linear procedures, under CP, DP and IP configurations.

	$\mathcal{XF}$ normal	$\mathcal{XF}$ 0-1	$\mathcal{XF}$ linear
CP	1441	1192	461
DP	1920	1561	827
IP	1186	994	322

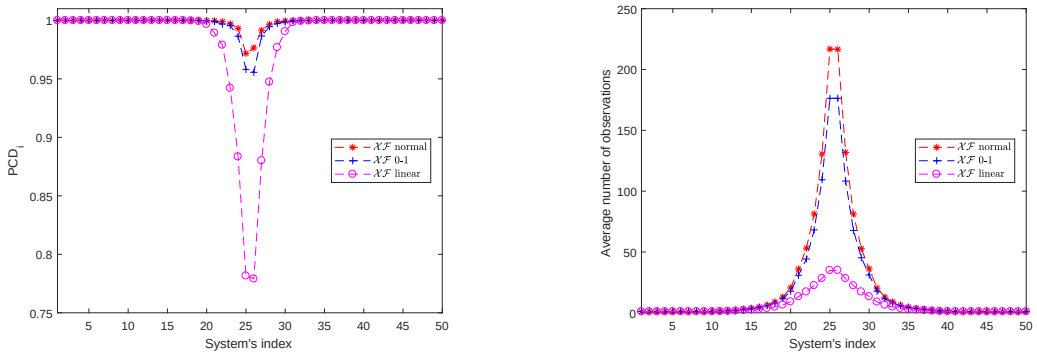
The next comparison is among the $\mathcal{XF}$ normal, $\mathcal{BK}$ and $\mathcal{GC}$ procedures. Given that the three procedures take approximately the same average total number of observations per replication (see Table 2.7), the $\mathcal{XF}$ normal procedure still performs the best among the three procedures in all three configurations in terms of PCD based on Table 2.6. Figure 2.2 again shows the advantage of the $\mathcal{XF}$ normal procedure on deciding feasibilities of the



(a) Constant Precisions



(b) Decreasing Precisions



(c) Increasing Precisions

Figure 2.1: PCD_i and average number of observations spent on each system on each system i by using the $\mathcal{XF}$ normal, $\mathcal{XF}$ 0-1 and $\mathcal{XF}$ linear procedures, under CP, DP and IP configurations.

barely feasible/infeasible systems. The $\mathcal{XF}$ normal procedure slightly outperforms the $\mathcal{GC}$ procedure in all three configurations, but significantly better than the $\mathcal{BK}$ procedure. In our experiments, we see that the $\mathcal{BK}$ is conservative, compared to the other two. In order to make the $\mathcal{BK}$ procedure have about the same average total numbers of observations as

the other two, we have to choose some values of $(1-\alpha)$ and ϵ that may not be reasonable in real applications. Consequently, we see that the performance of the $\mathcal{BK}$ procedure is compromised, especially in the DP configuration.

Table 2.6: Summary of PCD by using $\mathcal{XF}$ normal, $\mathcal{BK}$ and $\mathcal{GC}$, under CP, DP and IP configurations.

	$\mathcal{XF}$ normal	$\mathcal{BK}$	$\mathcal{GC}$
CP	0.9004	0.7575	0.8807
DP	0.8412	0.3145	0.8143
IP	0.9258	0.8927	0.9070

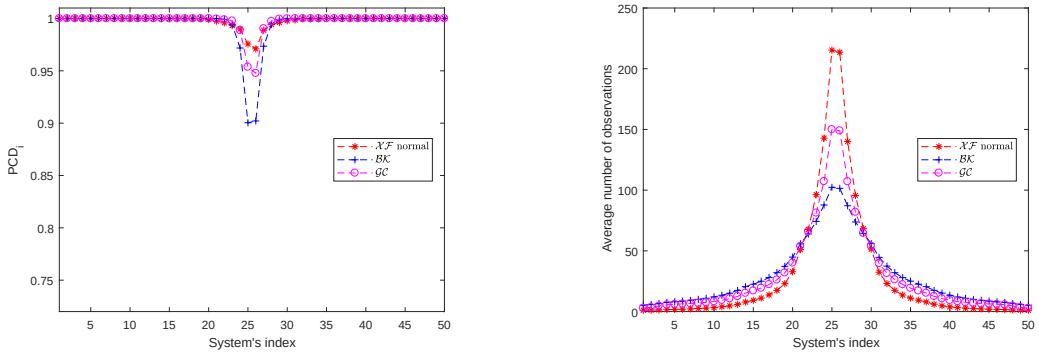
Table 2.7: Summary of average total number of observations taken by the $\mathcal{XF}$ normal, $\mathcal{BK}$ and $\mathcal{GC}$, under CP, DP and IP configurations.

	$\mathcal{XF}$ normal	$\mathcal{BK}$	$\mathcal{GC}$
CP	1441	1448	1441
DP	1920	1918	1920
IP	1186	1186	1186

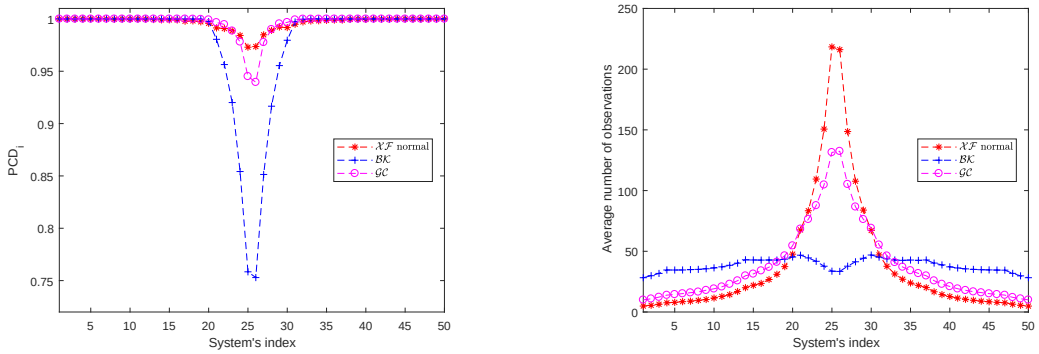
2.4 Conclusion

In this chapter, we introduce a new reward function, called normal reward function, that tends to assign more reward on barely feasible/infeasible systems than clearly feasible/infeasible ones. From our experiments, we see the advantage of using the normal reward function, while compared to the 0-1 and linear reward functions, in a Bayesian procedure to solve the feasibility determination problem. Our proposed procedure, $\mathcal{XF}$ normal, spends simulation effort wisely, as it takes very few observations on clearly feasible/infeasible systems and more on barely feasible/infeasible systems to improve the overall decision accuracy. The experimental results show that the $\mathcal{XF}$ normal procedure performs better than the other two types of procedures: the IZ-type $\mathcal{BK}$ and OCBA-type $\mathcal{GC}$ procedures.

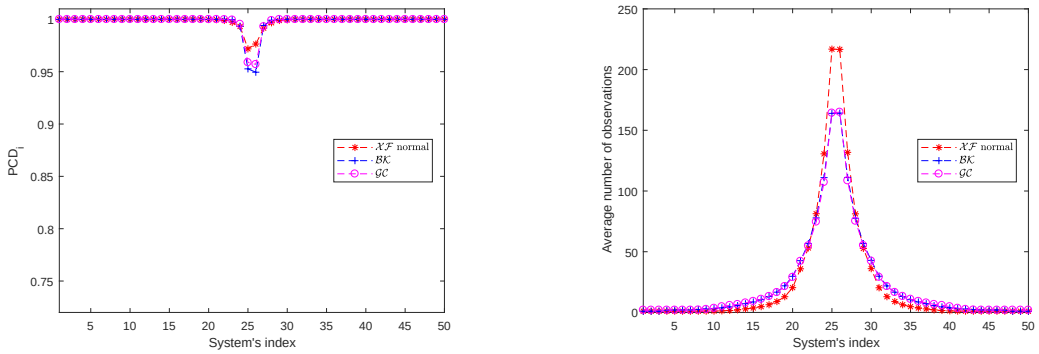
However, there are several drawbacks with the $\mathcal{XF}$ normal procedure. First of all, the procedure requires heavy calculation, which involves a backward algorithm and a state space search, in order to obtain the continuation region. Second, the procedure assumes



(a) Constant Precisions



(b) Decreasing Precisions



(c) Increasing Precisions

Figure 2.2: PCD_i and average number of observations spent on each system on each system i by using the $\mathcal{XF}$ normal, $\mathcal{BK}$ and $\mathcal{GC}$, under CP, DP and IP configurations.

that the sampling precision is known to decision maker, which is not true in most realistic scenarios. Lastly, the procedure can only work for cases where the unit sampling costs are given, but sometimes, it may be difficult to specify those costs. In Chapter 3, we develop a new type of Bayesian procedures for feasibility determination that can overcome the drawbacks mentioned above.

CHAPTER 3

NEW BAYESIAN SEQUENTIAL PROCEDURES FOR FEASIBILITY DETERMINATION

From previous chapters, we have seen that the problem of feasibility determination is present in many areas, and different types of statistical procedures have been developed to approach the problem. An indifference zone (IZ) type procedure provides a statistical guarantee on the selection quality such as the probability of correct decision (PCD). However, it is known to be conservative, as it usually takes more observations than actually needed in order to provide the guarantee. An optimal computing budget allocation (OCBA) procedure provides an asymptotically optimal budget allocation rule, but it requires that a total budget must be given ahead and does not provide any information about the quality of the decision. Lastly, a Bayesian type procedure allows a decision maker to choose a reward/loss function to describe the importance of making a correct decision, and utilizes the value of information (VOI) and Bayesian statistics to derive a Bayesian optimal sampling policy. However, a Bayesian procedure usually involves heavy computation. Each procedure in solving the feasibility determination problem has its own advantages and limitations. Consequently, the answer to which procedure is the best depends on the settings of the application and preference of the decision maker. From Chapter 2, we see that a Bayesian procedure can return promising performances. However, it also has several weaknesses that can be improved, which is exactly our motivation for this chapter.

In this chapter, we present new Bayesian procedures for feasibility determination that have several valuable characteristics. First, our procedures are flexible, as they can handle many different situations such as constrained or unconstrained total budget, small or large total budget, same or different unit cost for taking an observation, and two-stage or multiple-stages. Second, the procedures are easy for implementation and do not require

many input parameters. Lastly, the procedures have competitive performances in general and do not require heavy computation.

3.1 Background

In this section, we define notations and problem. As our work is built on almost the same background as Chapter 2, we continue to use the same notations from Chapter 2, but with some modifications to better formulate the problem.

3.1.1 Assumptions

As in Chapter 2, we consider a single constraint and assume that Assumptions 2 and 1 hold. However, one major difference is that we now assume that the sampling precision γ_i is unknown for each system i. The prior distributions then need to be updated accordingly. Using the conjugate priors on μ_i and γ_i, which are also called $normal - gamma$ priors, we have the following assumption:

Assumption 4. *For $i = 1, 2, \ldots, k$, $\gamma_i \sim \Gamma(\alpha_i, \lambda_i)$ and $\mu_i|_{\gamma_i} \sim \mathcal{N}(\eta_i, \gamma_i^{-1}/t_i)$, for some $t_i > 0$, where $\Gamma(\alpha, \lambda)$ is a gamma distribution with mean α/λ and variance α/λ^2.*

3.1.2 Problem Formulation

We consider there are k available systems and there is a fixed unit cost $c_i > 0$ associated with simulating an observation from system i. In addition, we let $\mathcal{B}$ be the budget constraint such that the total simulation effort is no larger than $\mathcal{B}$. Here, we formulate the problem by pretending that there are only two stages of taking observations. Later, we extend to sequential procedures based on the two-stage procedures.

Let $n_{0,i}$ be the sample size of first-stage observations for system i. By assuming a non-informative prior distribution and Assumption 4, it is straightforward (see [70]) to show

that after the first-stage observations,

$$\gamma_i \sim \Gamma\left((n_{0,i}-1)/2, \hat{\sigma}_i^2(n_{0,i}-1)/2\right),$$

$$\mu_i|_{\gamma_i} \sim \mathcal{N}\left(\bar{y}_i, \gamma_i^{-1}/n_{0,i}\right),$$

where $\bar{y}_i = \sum_{j=1}^{n_{0,i}} y_{ij}/n_{0,i}$ and $\hat{\sigma}_i^2 = \sum_{j=1}^{n_{0,i}} (y_{ij}-\bar{y}_i)^2/(n_{0,i}-1)$. Further, the marginal distribution for the unknown mean μ_i is described by a Student-t distribution:

$$\mu_i \sim St\left(\bar{y}_i, n_{0,i}/\hat{\sigma}_i^2, n_{0,i}-1\right),$$

where $St(\ell, g, v)$ is a Student-t distribution with mean ℓ, precision g and degree of freedom v. When $v > 2$, the variance of $St(\ell, g, v)$ is $\frac{g^{-1}v}{v-2}$.

Let n_i be the second-stage sample size and $\mathbf{y}_{n_i} = (y_{i,n_{0,i}+1}, y_{i,n_{0,i}+2}, \ldots, y_{i,n_{0,i}+n_i})$ be the second-stage observations from system i. Then, the posterior distributions of γ_i and μ_i given $\mathbf{y}_{n_i}$ are as follows:

$$\gamma_i|\mathbf{y}_{n_i} \sim \Gamma\left((n_{0,i}+n_i-1)/2, \hat{\bar{\sigma}}_i^2(n_{0,i}+n_i-1)/2\right),$$

$$\mu_i|\mathbf{y}_{n_i}, \gamma_i \sim N\left(\bar{\bar{y}}_i, \gamma_i^{-1}/(n_{0,i}+n_i)\right),$$

where $\bar{\bar{y}}_i = \sum_{j=1}^{n_{0,i}+n_i} y_{ij}/(n_{0,i}+n_i)$, $\hat{\bar{\sigma}}_i^2 = \sum_{j=1}^{n_{0,i}+n_i} (y_{ij}-\bar{\bar{y}}_i)^2/(n_{0,i}+n_i-1)$. The marginal distribution for μ_i given $\mathbf{y}_{n_i}$ is

$$\mu_i|\mathbf{y}_{n_i} \sim St\left(\bar{\bar{y}}_i, (n_{0,i}+n_i)/\hat{\bar{\sigma}}_i^2, n_{0,i}+n_i-1\right).$$

Therefore, we have that $\mathbb{E}\left[\mu_i|\mathbf{y}_{n_i}\right] = \bar{\bar{y}}_i$.

Our goal is to determine the sample size of second-stage observations n_i, after taking the first-stage observations. Consequently, the second-stage observations are actually un-

known and we denote them as variables $\mathbf{Y}_{n_i} = (Y_{i,n_{0,i}+1}, Y_{i,n_{0,i}+2}, \ldots, Y_{i,n_{0,i}+n_i})$. Then $\mathbb{E}\left[\mu_i|\mathbf{Y}_{n_i}\right] = \frac{n_{0,i}\bar{y}_i + n_i\bar{Y}_{n_i}}{n_{0,i}+n_i}$, where $\bar{Y}_{n_i} = \sum_{j=n_{0,i}+1}^{n_{0,i}+n_i} Y_{ij}/n_i$.

Since $\bar{Y}_{n_i}|\mu_i, \gamma_i \sim N\left(\mu_i, 1/(n_i\gamma_i)\right)$, we can find that

$$\bar{Y}_{n_i} \sim St\left(\bar{y}_i, \frac{1}{\hat{\sigma}_i^2}\frac{n_{0,i}n_i}{n_{0,i}+n_i}, n_{0,i}-1\right),$$

by integrating the joint distribution of $\bar{Y}_{n_i}$, μ_i and γ_i over all possible values of μ_i and γ_i. Eventually, it leads to

$$\mathbb{E}\left[\mu_i|\mathbf{Y}_{n_i}\right] \sim St\left(\bar{y}_i, \frac{1}{\hat{\sigma}_i^2}\frac{(n_{0,i}+n_i)n_{0,i}}{n_i}, n_{0,i}-1\right)$$

The loss function $L(\mu_i, D_i)$ is chosen by the decision maker to represent the loss when a system's mean performance is μ_i and the decision made on its feasibility is D_i. Here, we define $D_i = D^0$ as deciding the system as feasible, while $D_i = D^1$ as infeasible. Let $\mathbf{y_n} = (\mathbf{y}_{n_1}, \mathbf{y}_{n_2}, \ldots, \mathbf{y}_{n_k})$. In this chapter, we consider two loss functions: the 0-1 loss function and the normal loss function. Given all the second-stage observations $\mathbf{y_n}$, the expected loss is $\mathbb{E}_{\boldsymbol{\mu}|\mathbf{y_n}}\left[\sum_{i=1}^k L(\mu_i, D_i)|\mathbf{y_n}\right]$, where $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_k)$.

The goal is to determine the second-stage allocation $\mathbf{n} = (n_1, n_2, \ldots, n_k)$ that minimizes the expected total loss ρ, which can be described by

$$\rho(\mathbf{n}) = \mathbb{E}_{\mathbf{y_n}}\left[\mathbb{E}_{\boldsymbol{\mu}|\mathbf{y_n}}\left[\sum_{i=1}^k L(\mu_i, D_i)|\mathbf{y_n}\right]\right] + \mathbf{cn}^T, \tag{3.1}$$

where $\mathbf{c} = (c_1, c_2, \ldots, c_k)$.

In the case of an unconstrained budget, we have the following optimization problem:

$$\min_{\mathbf{n}} \rho(\mathbf{n}), \text{ s.t. } n_i \geq 0 \text{ for } i = 1, 2, \ldots, k.$$

In the case of a constrained budget $\mathcal{B}$, the optimization problem becomes:

$$\min_{\mathbf{n}} \rho(\mathbf{n}),\ \text{s.t. } \mathbf{c}\mathbf{n}^T = \mathcal{B} - \mathbf{c}\mathbf{n_0}^T,$$

where $\mathbf{n_0} = (n_{0,1}, n_{0,2}, \ldots, n_{0,k})$.

For reference, Table 3.1 summarizes notation used throughout this chapter.

Table 3.1: Summary of notation.

notation	meaning
k	total number of available systems
d	threshold for the constraint on the performance measure of each system
c_i	cost per simulation for each system i, $i = 1, \ldots, k$
$\mathbf{c}$	vector of costs per simulation, $\mathbf{c} = (c_1, \ldots, c_k)$
$n_{0,i}$	first-stage sample size for each system i, $i = 1, \ldots, k$
$\mathbf{n_0}$	vector of first-stage sample sizes $\mathbf{n_0} = (n_{0,1}, \ldots, n_{0,k})$
n_i	second-stage sample size for each system i, $i = 1, \ldots, k$
$\mathbf{n}$	vector of second-stage sample sizes $\mathbf{n} = (n_1, \ldots, n_k)$
$\bar{y}_i$	first-stage sample mean for each system i, $i = 1, \ldots, k$
$\bar{\bar{y}}_i$	overall sample mean from first and second stages for each system i, $i = 1, \ldots, k$
$\hat{\sigma}_i^2$	first-stage sample variance for each system i, $i = 1, \ldots, k$
$\hat{\bar{\sigma}}_i^2$	overall sample variance from first and second stages for each system i, $i = 1, \ldots, k$
$\mathbf{Y}_{n_i}$	vector variables of the second-stage observations from system i, $i = 1, \ldots, k$
$\mathbf{y}_{n_i}$	vector of realizations of $\mathbf{Y}_{n_i}$, $i = 1, \ldots, k$
$\mathbf{y_n}$	vector of all realizations of the second-stage observations, $\mathbf{y_n} = (\mathbf{y}_{n_1}, \mathbf{y}_{n_2}, \ldots, \mathbf{y}_{n_k})$
μ_i	unknown mean performance for system i, $i = 1, \ldots, k$
γ_i	unknown sampling precision for system i, $i = 1, \ldots, k$
D_i	decision on feasibility of system i, $i = 1, \ldots, k$
$\mathcal{B}$	total computing budget
ρ	total expected loss

3.2 New Two-Stage Procedures

In this section, we derive the two-stage procedures for 0-1 and normal loss functions. The idea behind our derivation is to find a convenient expression of expected total loss $\rho(\mathbf{n})$, based on which we can determine the optimal choice of second stage sample sizes $\mathbf{n}$.

3.2.1 Two-Stage Procedures: 0-1 Loss

First, consider the 0-1 loss function which is defined as follows:

$$L(\mu_i, D^0) = \begin{cases} 0, & \text{if } \mu_i \leq d; \\ 1, & \text{otherwise,} \end{cases}$$

$$L(\mu_i, D^1) = \begin{cases} 1, & \text{if } \mu_i \leq d; \\ 0, & \text{otherwise,} \end{cases}$$

where D^0 is to decide the system is feasible, and D^1 is to decide the system is infeasible.

To facilitate the computation of the optimal choice of n_i, we introduce the loss difference as follows:

$$L'(\mu_i, D_i) = \begin{cases} L(\mu_i, D_i) - L(\mu_i, D^0), & \text{if } \bar{y}_i \leq d; \\ L(\mu_i, D_i) - L(\mu_i, D^1), & \text{otherwise.} \end{cases} \tag{3.2}$$

The function $L'(\mu_i, D_i)$ can be understood as the difference of loss between the decision made after second stage and the one made after the first stage. For further derivation, we look at different situations one by one.

First, suppose that $\bar{y}_i \leq d$. Then

$$L'(\mu_i, D_i) = \begin{cases} 0, & \text{if } D_i = D^0; \\ 1, & \text{if } \mu \leq d \text{ and } D_i = D^1; \\ -1, & \text{if } \mu > d \text{ and } D_i = D^1. \end{cases}$$

The expected difference of loss with respect to $\mu_i|\mathbf{Y}_{n_i}$ is

$$\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)\right] = \begin{cases} 0, & \text{if } D_i = D^0; \\ 2\text{Pr}\left\{\mu_i \leq d|\mathbf{Y}_{n_i}\right\} - 1, & \text{if } D_i = D^1. \end{cases}$$

The corresponding Bayesian optimal decision is

$$D^* = \begin{cases} D^0, & \text{if Pr}\left\{\mu_i \leq d|\ \mathbf{Y}_{n_i}\right\} > 1/2; \\ D^1, & \text{if Pr}\left\{\mu_i \leq d|\ \mathbf{Y}_{n_i}\right\} < 1/2; \\ D^0 \text{ or } D^1, & \text{if Pr}\left\{\mu_i \leq d|\ \mathbf{Y}_{n_i}\right\} = 1/2. \end{cases}$$

Since $\mu_i|\mathbf{Y}_{n_i}$ has a Student-t distribution, which is symmetrical, we have that

$\text{Pr}\left\{\mu_i|\mathbf{Y}_{n_i} \leq d\right\} > 1/2$ is equivalent to $\mathbb{E}\left[\mu_i|\mathbf{Y}_{n_i}\right] < d$, which is also equivalent to $\frac{n_{0,i}\bar{y}_i + n_i\bar{Y}_{n_i}}{n_{0,i}+n_i} < d$. Therefore, the Bayesian optimal decision is equivalent to

$$D^* = \begin{cases} D^0, & \text{if } \frac{n_{0,i}\bar{y}_i + n_i\bar{Y}_{n_i}}{n_{0,i}+n_i} \leq d; \\ D^1, & \text{otherwise.} \end{cases}$$

The expected loss with respect to μ_i is

$$\begin{aligned} \mathbb{E}_{\mu_i}\left[L(\mu_i, D_i)\right] &= \mathbb{E}_{\mu_i}\left[L'(\mu_i, D_i)\right] + \mathbb{E}_{\mu_i}\left[L(\mu_i, D^0)\right] \\ &= \mathbb{E}_{\mathbf{Y}_{n_i}}\left[\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)|\mathbf{Y}_{n_i}\right]\right] + \mathbb{E}_{\mu_i}\left[L(\mu_i, D^0)\right]. \end{aligned}$$

Since $\mathbb{E}_{\mu_i}\left[L(\mu_i, D^0)\right]$ is not related to n_i, we can focus on

$$\begin{aligned}
\mathbb{E}_{\mathbf{Y}_{n_i}}\left[\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)|\mathbf{Y}_{n_i}\right]\right] &= \int_{\mathbf{y}_{n_i}\in\Upsilon}\left[2\Pr\left\{\mu_i \le d|\,\mathbf{y}_{n_i}\right\} - 1\right]\cdot f_{\mathbf{Y}_{n_i}}(\mathbf{y}_{n_i})d\mathbf{y}_{n_i} \\
&\approx \int_{\mathbf{y}_{n_i}\in\Upsilon} -1\cdot f_{\mathbf{Y}_{n_i}}(\mathbf{y}_{n_i})d\mathbf{y}_{n_i} \\
&= -1\cdot\Pr\left\{\mathbf{Y}_{n_i}\in\Upsilon\right\} \\
&= -1\cdot\Pr\left\{\mathbb{E}\left[\mu_i|\mathbf{Y}_{n_i}\right] > d\right\} \\
&= -1\cdot\left[1-\Phi_{n_{0,i}-1}\left((d-\bar{y}_i)\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{\frac{1}{2}}\right)\right] \\
&= \Phi_{n_{0,i}-1}\left(|d-\bar{y}_i|\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{\frac{1}{2}}\right) - 1, \qquad (3.3)
\end{aligned}$$

where $f_{\mathbf{Y}_{n_i}}(\cdot)$ is the probability density function of $\mathbf{Y}_{n_i}$, $\Upsilon = \left\{\mathbf{y}_{n_i} : \frac{n_{0,i}\bar{y}_i + n_i\bar{y}_{n_i}}{n_{0,i}+n_i} > d\right\}$, and $\Phi_v(\cdot)$ is the cumulative distribution function of standard Student-t distribution with degree of freedom v. Note that we use an approximation in the second line of (3.3): as $(n_{0,i}+n_i)\to\infty$ and given that $\mathbf{y}_{n_i}\in\Upsilon$, $\Pr\left\{\mu_i \le d|\mathbf{y}_{n_i}\right\}\to 0$. This is intuitive: after obtaining a lot of observations, if we see that the overall sample mean $\bar{\bar{y}}_i > d$, we are confident that the system's mean performance $\mu_i > d$ and the probability of making a wrong decision is close to zero.

For $\bar{y}_i > d$, similar reasoning can be applied and $\mathbb{E}_{\mathbf{Y}_{n_i}}\left[\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)|\mathbf{Y}_{n_i}\right]\right]$ has the same expression as (3.3).

Finally, the expected total loss, which includes the cost of taking observations, has the following expression:

$$\rho_{0-1}(n_i) \approx \Phi_{n_{0,i}-1}\left(|d-\bar{y}_i|\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{\frac{1}{2}}\right) - 1 + \mathbb{E}_{\mu_i}\left[L(\mu_i, D')\right] + n_i c_i, \qquad (3.4)$$

where $D' = D^0$ if $\bar{y}_i \le d$ and $D' = D^1$ if $\bar{y} > d$.

Unconstrained Budget:

Suppose that we have an unconstrained budget, then our goal is to find n_i that minimizes $\rho_{0-1}(n_i)$. Assuming that n_i is continuous, we take the derivative of (3.4) with respect to n_i:

$$\begin{aligned}\frac{\partial \rho_{0-1}(n_i)}{\partial n_i} &\approx \phi_{n_{0,i}-1}\left(|d-\bar{y}_i|\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{\frac{1}{2}}\right)|d-\bar{y}_i|\left(\frac{1}{2}\right)\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{-\frac{1}{2}}\cdot \\ &\quad \frac{n_{0,i}^2}{\hat{\sigma}_i^2}\left(-\frac{1}{n_i^2}\right)+c_i \\ &= -\frac{1}{2}\phi_{n_{0,i}-1}\left(|d-\bar{y}_i|\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{\frac{1}{2}}\right)|d-\bar{y}_i|\left[\frac{(n_{0,i}+n_i)n_{0,i}}{n_i}\right]^{\frac{3}{2}}\cdot \\ &\quad \frac{1}{\hat{\sigma}_i(n_{0,i}+n_i)^2}+c_i \\ &\approx -\frac{1}{2}\phi_{n_{0,i}-1}\left((d-\bar{y}_i)\frac{n_{0,i}^{\frac{1}{2}}}{\hat{\sigma}_i}\right)|d-\bar{y}_i|\frac{n_{0,i}^{\frac{3}{2}}}{\hat{\sigma}_i(n_{0,i}+n_i)^2}+c_i. \end{aligned} \tag{3.5}$$

Here, we use another approximation based on the fact that the precision of the posterior mean $\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i} \to n_{0,i}/\hat{\sigma}_i^2$ as $n_i \to \infty$.

Setting (3.5) to zero, we can solve for n_i as follows:

$$n_i = \left[\frac{n_{0,i}^{\frac{3}{2}}}{2c_i\hat{\sigma}_i}|d-\bar{y}_i|\phi_{n_{0,i}-1}\left((d-\bar{y}_i)\frac{n_{0,i}^{\frac{1}{2}}}{\hat{\sigma}_i}\right)\right]^{\frac{1}{2}} - n_{0,i}. \tag{3.6}$$

Since some n_i returned by (3.6) may be negative, we need to remedy the non-negativity violation by setting those n_i to zero. Eventually, the asymptotic optimal choice of n_i is

$$n_i^* = \max\left\{\left[\frac{n_{0,i}^{\frac{3}{2}}}{2c_i\hat{\sigma}_i}|d-\bar{y}_i|\phi_{n_{0,i}-1}\left((d-\bar{y}_i)\frac{n_{0,i}^{\frac{1}{2}}}{\hat{\sigma}_i}\right)\right]^{\frac{1}{2}} - n_{0,i}, 0\right\}.$$

Constrained Budget:

Now, consider that there is a budget constraint $\mathbf{c}\mathbf{n}^T = \mathcal{B} - \mathbf{c}\mathbf{n_0}^T$. Let θ be a Lagrange multiplier for the constraint $\mathbf{c}\mathbf{n}^T = \mathcal{B} - \mathbf{c}\mathbf{n_0}^T$. Then, the asymptotic optimality condition

is

$$\theta c_i = -\frac{1}{2}\phi_{n_{0,i}-1}\left((d-\bar{y}_i)\frac{n_{0,i}^{\frac{1}{2}}}{\hat{\sigma}_i}\right)|d-\bar{y}_i|\frac{n_{0,i}^{\frac{3}{2}}}{\hat{\sigma}_i(n_{0,i}+n_i)^2}.$$

Therefore, we get

$$\frac{(n_{0,i}+n_i)^2}{(n_{0,j}+n_j)^2} = \frac{\phi_{n_{0,i}-1}\left((d-\bar{y}_i)\frac{n_{0,i}^{\frac{1}{2}}}{\hat{\sigma}_i}\right)|d-\bar{y}_i|\frac{n_{0,i}^{\frac{3}{2}}}{\hat{\sigma}_i c_i}}{\phi_{n_{0,j}-1}\left((d-\bar{y}_j)\frac{n_{0,j}^{\frac{1}{2}}}{\hat{\sigma}_j}\right)|d-\bar{y}_j|\frac{n_{0,j}^{\frac{3}{2}}}{\hat{\sigma}_j c_j}},$$

for any $i, j = 1, 2, \ldots, k$. Define $Q_i = \phi_{n_{0,i}-1}\left((d-\bar{y}_i)\frac{n_{0,i}^{\frac{1}{2}}}{\hat{\sigma}_i}\right)|d-\bar{y}_i|\frac{n_{0,i}^{\frac{3}{2}}}{\hat{\sigma}_i c_i}$. Then

$$n_j = \left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}}(n_{0,i}+n_i) - n_{0,j}.$$

From $\mathbf{cn}^T = \mathcal{B} - \mathbf{cn_0}^T$, we get

$$\sum_{j=1}^{k} n_j c_j = \sum_{j=1}^{k}\left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}}(n_{0,i}+n_i)c_j - \mathbf{cn_0}^T = \mathcal{B} - \mathbf{cn_0}^T,$$

and consequently,

$$n_i = \frac{\mathcal{B}}{\sum_{j=1}^{k}\left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}} c_j} - n_{0,i}. \tag{3.7}$$

Note that some n_i returned by (3.7) may be negative. We need to fix the non-negativity constraint violation. Specifically, let $K = \{1, 2, \ldots, k\}$. For each $i \in K$, remove i from K if n_i returned by (3.7) is negative and set $n_i = 0$. Then we can recompute the remaining n_i for each $i \in K$. From $\mathbf{cn}^T = \mathcal{B} - \mathbf{cn_0}^T$, we get

$$\sum_{j\in K} n_j c_j = \sum_{j\in K}\left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}}(n_{0,i}+n_i)c_j - \sum_{j\in K} n_{0,j}c_j = \mathcal{B} - \mathbf{cn_0}^T.$$

Solving for n_i, we have

$$n_i = \frac{\mathcal{B} - \mathbf{cn_0}^T + \sum_{j \in K} n_{0,j} c_j}{\sum_{j \in K} \left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}} c_j} - n_{0,i}. \tag{3.8}$$

For each $i \in K$, remove i from K if n_i returned by (3.8) is negative, set $n_i = 0$ and recompute the remaining n_i for each $i \in K$ using (3.8). The process needs to be repeated until n_i returned by (3.8) are nonnegative for all $i \in K$.

Eventually, the asymptotic optimal choice of n_i with a budget constraint is

$$n_i^* = \begin{cases} \frac{\mathcal{B} - \mathbf{cn_0}^T + \sum_{j \in K} n_{0,j} c_j}{\sum_{j \in K} \left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}} c_j} - n_{0,i}, & \text{if } i \in K; \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

3.2.2 Two-Stage Procedures: Normal Loss

In this section, we derive the sampling rule for the normal loss. The normal loss function is defined as follows:

$$L(\mu_i, D^0) = \begin{cases} 0, & \text{if } \mu_i \leq d; \\ a \cdot \exp\{-\frac{1}{2}(\mu_i - d)^2 \cdot b\}, & \text{otherwise;} \end{cases}$$

$$L(\mu_i, D^1) = \begin{cases} a \cdot \exp\{-\frac{1}{2}(\mu_i - d)^2 \cdot b\}, & \text{if } \mu_i \leq d; \\ 0, & \text{otherwise,} \end{cases}$$

where D^0 is to decide the system is feasible, and D^1 is to decide the system is infeasible.

Define the loss difference $L'(\mu_i, D_i)$ as (3.2), and suppose that $\bar{y}_i \leq d$. Then

$$L'(\mu_i, D_i) = \begin{cases} 0, & \text{if } D_i = D^0; \\ a \cdot \exp\{-\frac{1}{2}(\mu_i - d)^2 \cdot b\}, & \text{if } \mu_i \leq d \text{ and } D_i = D^1; \\ -a \cdot \exp\{-\frac{1}{2}(\mu_i - d)^2 \cdot b\}, & \text{if } \mu_i > d \text{ and } D_i = D^1. \end{cases}$$

Due to the difficulty of finding the expected loss when assuming γ_i is unknown, the following derivation is under the assumption that γ_i is known. Then, we have the following distributions:

$$\begin{aligned} &\bar{Y}_{n_i}|\mu_i \sim \mathcal{N}\left(\mu_i, \gamma_i^{-1}/n_i\right), \\ &\mu_i \sim \mathcal{N}\left(\bar{y}_i, \gamma_i^{-1}/n_{0,i}\right), \\ &\bar{Y}_{n_i} \sim \mathcal{N}\left(\bar{y}_i, \frac{(n_{0,i}+n_i)\gamma_i^{-1}}{n_{0,i}n_i}\right), \\ &\mu_i|\mathbf{Y}_{n_i} \sim \mathcal{N}\left(\frac{n_{0,i}\bar{y}_i + n_i\bar{Y}_{n_i}}{n_{0,i}+n_i}, \frac{\gamma_i^{-1}}{(n_{0,i}+n_i)}\right). \end{aligned}$$

Let $\eta_i = \frac{n_{0,i}\bar{y}_i + n_i\bar{Y}_{n_i}}{n_{0,i}+n_i}$ and $\lambda_i = (n_{0,i}+n_i)\gamma_i$. Then $\eta_i \sim N\left(\bar{y}_i, \frac{n_i}{n_{0,i}}{\lambda_i}^{-1}\right)$.

As $\mu_i|\mathbf{Y}_{n_i} \sim N\left(\eta_i, \lambda_i^{-1}\right)$, we get

$$\begin{aligned} &\int_{-\infty}^{d} a \cdot \exp\left\{-\frac{1}{2}(d-\mu_i)^2 b\right\} \cdot \frac{\sqrt{\lambda_i}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\mu_i - \eta_i)^2\lambda_i\right\} d\mu_i \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}} \int_{-\infty}^{d} \frac{\sqrt{b}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(d-\mu_i)^2 b\right\} \cdot \frac{\sqrt{\lambda_i}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\mu_i - \eta_i)^2\lambda_i\right\} d\mu_i \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta_i, b, \lambda_i) \int_{-\infty}^{d} \frac{1}{\sqrt{2\pi}} \sqrt{b+\lambda_i} \exp\left\{-\frac{1}{2}(\mu_i - \frac{db+\eta_i\lambda_i}{b+\lambda_i})^2 (b+\lambda_i)\right\} d\mu_i \\ &= \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta_i, b, \lambda_i) \cdot \Phi\left(\left(d - \frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right), \end{aligned}$$

where $G(\eta_f, \eta_g, \lambda_f, \lambda_g) = \frac{1}{\sqrt{2\pi}}\sqrt{\frac{\lambda_f\lambda_g}{\lambda_f+\lambda_g}} \exp\left\{-\frac{1}{2}(\eta_f - \eta_g)^2 \frac{\lambda_f\lambda_g}{\lambda_f+\lambda_g}\right\}$, and $\Phi(\cdot)$ is the cumu-

lative distribution function of a standard normal distribution. Using similar reasoning,

$$\begin{aligned}
&\int_d^{\infty} a \cdot \exp\left\{-\frac{1}{2}(d-\mu_i)^2 b\right\} \cdot \frac{\sqrt{\lambda_i}}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\mu_i-\eta_i)^2\lambda_i\right\} d\mu_i \\
&= \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d,\eta_i,b,\lambda_i) \cdot \left[1-\Phi\left(\left(d-\frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right)\right].
\end{aligned}$$

The corresponding Bayesian optimal decision is

$$D^* = \begin{cases} D^0, & \text{if } \Phi\left(\left(d-\frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right) > 1/2; \\ D^1, & \text{if } \Phi\left(\left(d-\frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right) < 1/2; \\ D^0 \text{ or } D^1, & \text{otherwise.} \end{cases}$$

Since $\Phi\left(\left(d-\frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right) \geq 1/2$ is equivalent to $d \geq \eta_i$, the Bayesian optimal decision is equivalent to

$$D^* = \begin{cases} D^0, & \text{if } \frac{n_{0,i}\bar{y}_i + n_i \bar{Y}_{n_i}}{n_{0,i}+n_i} \leq d; \\ D^1, & \text{otherwise.} \end{cases}$$

We can compute $\mathbb{E}_{\mathbf{Y}_{n_i}}\left[\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)|\mathbf{Y}_{n_i}\right]\right]$ as follows:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Y}_{n_i}}\left[\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)|\mathbf{Y}_{n_i}\right]\right] \\
&= \int_{\mathbf{y}_{n_i}\in\Upsilon} \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta_i, b, \lambda_i) \\
&\quad \times \left[2\Phi\left(\left(d - \frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right) - 1\right] \times f_{\mathbf{Y}_{n_i}}(\mathbf{y}_{n_i}) d\mathbf{y}_{n_i} \\
&\approx \int_{\mathbf{y}_{n_i}\in\Upsilon} -\frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta_i, b, \lambda_i) \times f_{\mathbf{Y}_{n_i}}(\mathbf{y}_{n_i}) d\mathbf{y}_{n_i} \\
&= \int_{d}^{\infty} -\frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta_i, b, \lambda_i) \times f_{\eta_i}(\eta_i) d\eta_i \\
&= \int_{d}^{\infty} -\frac{a\sqrt{2\pi}}{\sqrt{b}} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b\lambda_i}{b+\lambda_i}} \exp\left\{-\frac{1}{2}(d-\eta_i)^2 \frac{b\lambda_i}{b+\lambda_i}\right\} \\
&\quad \times \frac{1}{\sqrt{2\pi}} \sqrt{\frac{n_{0,i}\lambda_i}{n_i}} \exp\left\{\frac{1}{2}(\eta_i - \bar{y}_i)^2 \frac{n_{0,i}\lambda_i}{n_i}\right\} d\eta_i \\
&= \frac{a\sqrt{2\pi}}{\sqrt{b}} G\left(d, \bar{y}_i, \frac{b\lambda_i}{b+\lambda_i}, \frac{n_{0,i}\lambda_i}{n_i}\right) \times \left[\Phi\left(|d - w_i|\sqrt{u_i}\right) - 1\right], \qquad (3.10)
\end{aligned}
$$

where $f_{\eta_i}(\cdot)$ is the probability density function of η_i, $\Upsilon = \left\{\mathbf{y}_{n_i} : \frac{n_{0,i}\bar{y}_i + n_i\bar{y}_{n_i}}{n_{0,i}+n_i} > d\right\}$, $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution, $w_i = \frac{d\frac{b\lambda_i}{b+\lambda_i} + \bar{y}_i\frac{n_{0,i}\lambda_i}{n_i}}{\frac{b\lambda_i}{b+\lambda_i} + \frac{n_{0,i}\lambda_i}{n_i}}$, $u_i = \frac{b\lambda_i}{b+\lambda_i} + \frac{n_{0,i}\lambda_i}{n_i}$. Note that we use an approximation in (3.10): as $(n_i + n_{0,i}) \to \infty$, $\lambda_i \to \infty$, and consequently, $\Phi\left(\left(d - \frac{db+\eta_i\lambda_i}{b+\lambda_i}\right)\sqrt{b+\lambda_i}\right) \to 0$, given that $\mathbf{y}_{n_i} \in \Upsilon$.

For $\bar{y}_i > d$, similar reasoning can be applied and $\mathbb{E}_{\mathbf{Y}_{n_i}}\left[\mathbb{E}_{\mu_i|\mathbf{Y}_{n_i}}\left[L'(\mu_i, D_i)|\mathbf{Y}_{n_i}\right]\right]$ has the same expression as (3.10).

Let $\rho_{normal}(n_i)$ be the total expected loss with respect to the normal loss function. Then $\rho_{normal}(n_i)$ has the following expression:

$$
\begin{aligned}
\rho_{normal}(n_i) \approx \frac{a\sqrt{2\pi}}{\sqrt{b}} G\left(d, \bar{y}_i, \frac{b\lambda_i}{b+\lambda_i}, \frac{n_{0,i}\lambda_i}{n_i}\right) \cdot \left[\Phi\left(|d - w_i|\sqrt{u_i}\right) - 1\right] \\
+ \mathbb{E}_{\mu_i}\left[L(\mu_i, D')\right] + n_i c_i, \qquad (3.11)
\end{aligned}
$$

where $D' = D^0$ if $\bar{y}_i \leq d$ and $D' = D^1$ if $\bar{y}_i > d$.

Unconstrained Budget:

Suppose that there is an unconstrained budget. Simple calculation shows that

$$G\left(d, \bar{y}_i, \frac{b\lambda_i}{b+\lambda_i}, \frac{n_{0,i}\lambda_i}{n_i}\right) = \left(\frac{bn_{0,i}}{b\gamma_i^{-1}+n_{0,i}}\right)^{\frac{1}{2}} \phi\left((d-\bar{y}_i)\left(\frac{bn_{0,i}}{b\gamma_i^{-1}+n_{0,i}}\right)^{\frac{1}{2}}\right),$$

which is not dependent on n_i. Therefore, the derivative of (3.11) with respect to n_i is

$$\frac{\partial \rho_{normal}(n_i)}{\partial n_i} \approx \frac{a\sqrt{2\pi}}{\sqrt{b}} G\left(d, \bar{y}_i, \frac{b\lambda_i}{b+\lambda_i}, \frac{n_{0,i}\lambda_i}{n_i}\right) \cdot \phi\left((d-w_i)\sqrt{u_i}\right) \cdot \frac{\partial\left[(d-w_i)\sqrt{u_i}\right]}{\partial n_i} + c_i.$$

Furthermore, we have

$$\frac{\partial(d-w_i)}{\partial n_i} = -(d-\bar{y}_i) \times \frac{n_{0,i} b\gamma_i^{-1}}{(n_{0,i}+n_i)^2(b\gamma_i^{-1}+n_{0,i})},$$

and

$$\frac{\partial u_i}{\partial n_i} = \frac{b^2\gamma_i^{-1}}{\left(b^2\gamma_i^{-1}+n_{0,i}+n_i\right)^2} - \frac{n_{0,i}^2\gamma_i}{n_i^2}.$$

Consequently,

$$\begin{aligned}
\frac{\partial\left[(d-w_i)\sqrt{u_i}\right]}{\partial n_i} &= -(d-\bar{y}_i) \times \frac{n_{0,i} b\gamma_i^{-1}}{(n_{0,i}+n_i)^2(b\gamma_i^{-1}+n_{0,i})}\sqrt{u_i} \\
&\quad + (d-w_i) \times \frac{1}{2} u_i^{-\frac{1}{2}} \left[\frac{b^2\gamma_i^{-1}}{\left(b^2\gamma_i^{-1}+n_{0,i}+n_i\right)^2} - \frac{n_{0,i}^2\gamma_i}{n_i^2}\right] \\
&\approx -(d-\bar{y}_i) \times \frac{n_{0,i} b\gamma_i^{-1}}{(n_{0,i}+n_i)^2(b\gamma_i^{-1}+n_{0,i})}\sqrt{\tilde{u}_i},
\end{aligned}$$

where $\tilde{u}_i = b + n_{0,i}\gamma_i$.

Putting all pieces together, we have

$$\begin{aligned}
\frac{\partial \rho_{normal}(n_i)}{\partial n_i} \approx -\frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot \left(\frac{bn_{0,i}}{b\gamma_i^{-1}+n_{0,i}}\right)^{\frac{1}{2}} \phi\left((d-\bar{y}_i)\left(\frac{bn_{0,i}}{b\gamma_i^{-1}+n_{0,i}}\right)^{\frac{1}{2}}\right) \\
\times \phi\left((d-\tilde{w}_i)\sqrt{\tilde{u}_i}\right)(d-\bar{y}_i)\frac{n_{0,i} b\gamma_i^{-1}}{(n_{0,i}+n_i)^2(b\gamma_i^{-1}+n_{0,i})}\sqrt{\tilde{u}_i} + c_i, \quad (3.12)
\end{aligned}$$

where $\tilde{w}_i = \frac{db+n_{0,i}\bar{y}\gamma_i}{b+n_{0,i}\gamma_i}$.

Setting (3.12) to zero, we get n_i as follows:

$$\begin{aligned} n_i = \Bigg[\frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \phi \left((d - \bar{y}_i) \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \right) \\ \times \phi \left((d - \tilde{w}_i)\sqrt{\tilde{u}_i} \right) (d - \bar{y}_i) \frac{n_{0,i} b\gamma_i^{-1}}{c_i(b\gamma_i^{-1} + n_{0,i})} \sqrt{\tilde{u}_i} \Bigg]^{\frac{1}{2}} - n_{0,i}. \end{aligned} \tag{3.13}$$

If there is any $n_i < 0$, we need to set it to zero because of the non-negativity constraint. Therefore, the asymptotically optimal choice is n_i is

$$\begin{aligned} n_i^* = \max \Bigg\{ \Bigg[\frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \phi \left((d - \bar{y}_i) \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \right) \\ \times \phi \left((d - \tilde{w}_i)\sqrt{\tilde{u}_i} \right) (d - \bar{y}_i) \frac{n_{0,i} b\gamma_i^{-1}}{c_i(b\gamma_i^{-1} + n_{0,i})} \sqrt{\tilde{u}_i} \Bigg]^{\frac{1}{2}} - n_{0,i}, 0 \Bigg\}. \end{aligned} \tag{3.14}$$

Constrained Budget:

Now suppose that there is a budget constraint $\mathbf{cn}^T = \mathcal{B} - \mathbf{cn_0}^T$. Let θ be a Lagrange multiplier for the constraint $\mathbf{cn}^T = \mathcal{B} - \mathbf{cn_0}^T$. Then, the asymptotic optimality condition is

$$\begin{aligned} \theta c_i = -\frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \phi \left((d - \bar{y}_i) \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \right) \\ \times \phi \left((d - \tilde{w}_i)\sqrt{\tilde{u}_i} \right) (d - \bar{y}_i) \frac{n_{0,i} b\gamma_i^{-1}}{(n_{0,i} + n_i)^2(b\gamma_i^{-1} + n_{0,i})} \sqrt{\tilde{u}_i}. \end{aligned}$$

Define Q_i be as follows:

$$\begin{aligned} Q_i = \frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \phi \left((d - \bar{y}_i) \left(\frac{bn_{0,i}}{b\gamma_i^{-1} + n_{0,i}} \right)^{\frac{1}{2}} \right) \\ \times \phi \left((d - \tilde{w}_i)\sqrt{\tilde{u}_i} \right) (d - \bar{y}_i) \frac{n_{0,i} b\gamma_i^{-1}}{(b\gamma_i^{-1} + n_{0,i})c_i} \sqrt{\tilde{u}_i}. \end{aligned}$$

Then, we have

$$\frac{(n_{0,i}+n_i)^2}{(n_{0,j}+n_j)^2}=\frac{Q_i}{Q_j},$$

for any $i,j=1,2,\ldots,k$.

Using similar reasoning as with the 0-1 loss function, we get

$$n_i=\frac{\mathcal{B}}{\sum_{j=1}^{k}\left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}}c_j}-n_{0,i}. \tag{3.15}$$

Again, some n_i returned by (3.15) may be negative. Here, we apply the same method to fix the non-negativity constraint violation as with the 0-1 loss function. Let $K=\{1,2,\ldots,k\}$. For each $i\in K$, remove i from K if n_i returned by (3.15) is negative and set $n_i=0$. Then recompute the remaining n_i for each $i\in K$ as follows:

$$n_i=\frac{\mathcal{B}-\mathbf{cn_0}^T+\sum_{j\in K}n_{0,j}c_j}{\sum_{j\in K}\left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}}c_j}-n_{0,i}. \tag{3.16}$$

For each $i\in K$, remove i from K if n_i returned by (3.16) is negative, set $n_i=0$ and recompute the remaining n_i for each $i\in K$ using (3.16). We repeat this process until n_i returned by (3.16) are nonnegative for all $i\in K$.

Eventually, the asymptotic optimal choice of n_i with a budget constraint is

$$n_i^*=\begin{cases}\frac{\mathcal{B}-\mathbf{cn_0}^T+\sum_{j\in K}n_{0,j}c_j}{\sum_{j\in K}\left(\frac{Q_j}{Q_i}\right)^{\frac{1}{2}}c_j}-n_{0,i}, & \text{if } i\in K;\\ 0, & \text{otherwise.}\end{cases} \tag{3.17}$$

Since two-stage procedures assign all remaining budget based on information from first-stage observations only, which is much more risky than sequential ones, the performances of two-stage procedures are usually inferior to those of sequential procedures. Therefore, we don't conduct further investigations on the performances of the new two-stage procedures presented above, but focus on developing new sequential procedures using the results

from the two-stage procedures.

3.3 New Sequential Procedures

We develop the sequential procedures by assigning an incremental budget Δ and allocating it to a single system that minimizes the expected loss for taking observations from the next stage, until a total budget $\mathcal{B}$ is spent. More specifically, the optimization problem to be solved at each stage becomes:

$$\min_i \rho(\mathbf{n}), \text{ s.t. } c_i n_i = \Delta \text{ and } n_j = 0 \text{ for all } j \neq i,$$

where Δ is the incremental budget at each stage. In general, the idea is that at each stage, we assume that there is only one stage left with a budget Δ. Since Δ is assumed to be small, it is reasonable to assign it to a single system only, and this system should be chosen to minimize the expected loss for spending the budget Δ. Then, this process is repeated at each stage until a total budget $\mathcal{B}$ is spent. We understand that our sequential procedures are myopic, as they try to minimize the expected loss from one stage at a time, rather than minimizing the overall expected loss throughout all stages. However, in return for being myopic, the procedures require much lighter computational efforts than procedures considering all stages at once, such as those in [33].

3.3.1 Sequential Procedures: 0-1 Loss

Since the asymptotic allocation rule from (3.9) relies on the assumption that the incremental budget Δ is large, which is not true in many applications, it may not be appropriate to directly apply (3.9) in sequential procedures. For a small Δ, we propose a procedure, denoted as $\mathcal{NB}_{0-1}(M)$, that takes observations from only one system to minimize the expected loss at each stage. From (3.4), we have that the difference in expected loss when

system i is sampled is

$$\begin{aligned}\rho'_{0-1}(n_i) &= \rho_{0-1}(n_i) - \mathbb{E}_{\mu_i}\left[L(\mu_i, D')\right] \\ &\approx \Phi_{n_{0,i}-1}\left(|d-\bar{y}_i|\left[\frac{(n_{0,i}+n_i)n_{0,i}}{\hat{\sigma}_i^2 n_i}\right]^{\frac{1}{2}}\right) - 1 + n_i c_i. \end{aligned} \tag{3.18}$$

The approximation in (3.18) holds, because we assume that the next stage is the last and therefore, $n_{0,i}$ is sufficiently large. Under the assumption that only one system can be sampled at each stage, the optimal choice of system is the one that minimizes (3.18). The $\mathcal{NB}_{0-1}(M)$ procedure is given in Algorithm 2.

Algorithm 2 $\mathcal{NB}_{0-1}(M)$

1: **Setup**: Specify total simulation budget $\mathcal{B}$, unit costs $\mathbf{c}$, initial simulation replication number n_0, incremental budget Δ_0, and threshold d.
2: **Initialization**: For each system i, obtain n_0 observations y_{ij}, $j = 1, 2, \ldots, n_0$. Set $n_{0,i} = n_0$ for all i and total used budget as $\Delta_T = \mathbf{c}\mathbf{n_0}^T$. Compute $\bar{y}_i$ and $\hat{\sigma}_i^2$ for all i.
3: **Iteration**: If $\Delta_T = \mathcal{B}$, stop and go to **Decision**. Otherwise, provide an incremental simulation budget $\Delta = \min\{\Delta_0, \mathcal{B} - \Delta_T\}$. For each system i, compute the difference in expected loss $\rho'_{0-1}(n_i)$ using (3.18) with $n_i = \Delta$. Choose $i^* = \arg\min\{\rho'_{0-1}(n_i)\}$. Take additional observations Δ from system i^*. Set $n_{0,i} = n_{0,i} + \Delta$. Update Δ_T, $\bar{y}_i$ and $\hat{\sigma}_i^2$.
4: **Decision**: Return $F = \{i : \bar{y}_i < d\}$ as the feasible set.

3.3.2 Sequential Procedures: Normal Loss

With the normal loss function, the sequential procedures are similar as the ones with the 0-1 loss function. However, since we assume that the sampling precision γ_i is known in our derivation but it is actually unknown in the application, we need to estimate γ_i. We use the maximum likelihood estimator of γ_i, which is $1/\hat{\sigma}_i^2$, to estimate γ_i.

Similar to the 0-1 loss, we propose a procedure, denoted as $\mathcal{NB}_{normal}(M)$, that assumes Δ is small and takes observations from only one system at each stage. The $\mathcal{NB}_{normal}(M)$

is based on the difference in expected loss, which is computed as

$$\begin{aligned}\rho'_{normal}(n_i) &= \rho_{normal}(n_i) - \mathbb{E}_{\mu_i}\left[L(\mu_i, D')\right] \\ &\approx \frac{a\sqrt{2\pi}}{\sqrt{b}} G\left(d, \bar{y}_i, \frac{b\lambda_i}{b+\lambda_i}, \frac{n_{0,i}\lambda_i}{n_i}\right) \cdot \left[\Phi\left(|d-w_i|\sqrt{u_i}\right) - 1\right] + n_i c_i, \quad (3.19)\end{aligned}$$

using the result from (3.11). Note that the approximation holds as $n_{0,i} \to \infty$. The $\mathcal{NB}_{normal}(M)$ procedure is given in Algorithm 3.

Algorithm 3 $\mathcal{NB}_{normal}(M)$

1: **Setup**: Specify total simulation budget $\mathcal{B}$, unit costs $\mathbf{c}$, initial simulation replication number n_0, incremental budget Δ_0, and threshold d.
2: **Initialization**: For each system i, obtain n_0 observations y_{ij}, $j = 1, 2, \ldots, n_0$. Set $n_{0,i} = n_0$ for all i and total used budget as $\Delta_T = \sum_{i=1}^{k} c_i n_{0,i}$. Compute $\bar{y}_i$ and $\hat{\sigma}_i^2$ for all i. Set $\gamma_i = 1/\hat{\sigma}_i^2$.
3: **Iteration**: If $\Delta_T = \mathcal{B}$, stop and go to **Decision**. Otherwise, provide an incremental simulation budget $\Delta = \min\{\Delta_0, \mathcal{B} - \Delta_T\}$. For each system i, compute the difference in expected loss $\rho'_{normal}(n_i)$ using (3.19) with $n_i = \Delta$. Choose $i^* = \arg\min\{\rho'_{normal}(n_i)\}$. Take additional observations Δ from system i^*. Set $n_{0,i} = n_{0,i} + n_i^{(r)}$. Update Δ_T, $\bar{y}_i$, $\hat{\sigma}_i^2$ and γ_i.
4: **Decision**: Return $F = \{i : \bar{y}_i < d\}$ as the feasible set.

3.4 Numerical Experiments

In this section, we compare the performances of procedures $\mathcal{NB}_{0-1}(M)$, $\mathcal{NB}_{normal}(M)$ and $\mathcal{GC}$ by numerical experiments. Since our proposed sequential procedures require a total budget constraint and [32] reports that the OCBA-type procedure $\mathcal{GC}$ outperforms the IZ-type $\mathcal{BK}$ procedure with a given total budget, we don't present comparisons of our procedures against the $\mathcal{BK}$ procedure here.

3.4.1 Experimental Settings

We consider $k = 20$ systems. We set the threshold $d = 0$ and unit cost $c_i = 1$ for all systems $i = 1, 2, \ldots, 20$. In addition, we set the default incremental budget $\Delta_0 = 10$. For

sampling precisions, we set $\gamma_i = 0.5$ for all i. We vary the total budget $\mathcal{B}$ from 210 to 3000. For the $\mathcal{NB}_{normal}(M)$ procedure, we set the parameters $a = 1$ and $b = \sum_{i=1}^{20} (1/\hat{\sigma}_i^2)\,/20$.

We consider three configurations for the systems' mean performances:

- The slippage configuration (SC), where the systems' mean performances are either δ better than d or δ worse than d. For feasible systems ($i = 1, 2, \ldots, 10$), the mean performances are $\mu_i = -0.3$, while for infeasible systems ($i = 11, 12, \ldots, 20$), $\mu_i = 0.3$.

- The constant difference configuration (CDC), where any two consecutive systems' mean performances are δ apart from each other. Specifically, $\mu_i = -2.1 + 0.2 \cdot i$, for $i = 1, 2, \ldots, 20$.

- The normal random configuration (NRC), where the systems' mean performances are randomly selected from a normal distribution. Specifically, we choose $\mu_i \sim \mathcal{N}(0, 2)$ for $i = 1, 2, \ldots, 20$.

3.4.2 Results

To compare the performances of the procedures, we look at three different metrics: PCD, expected 0-1 loss and expected normal loss. The definition of PCD is the same as in Chapter 2, which is the probability of making correct decisions on feasibilities of all available systems. The expected 0-1 loss and expected normal loss are the expectations of the 0-1 loss and normal loss, respectively. For each procedure, we estimate its PCD, expected 0-1 loss and expected normal loss by averaging results from 10000 independent replications.

From Figures 3.1, 3.2 and 3.3, we see that the performances of the three procedures are quite consistent throughout the three configurations. The results show that the $\mathcal{NB}_{0-1}(M)$ and $\mathcal{NB}_{normal}(M)$ procedures both outperform the $\mathcal{GC}$ procedure in terms of PCD, 0-1 loss and normal loss, in all three configurations. We observe that the $\mathcal{NB}_{0-1}(M)$ and $\mathcal{NB}_{normal}(M)$ procedures have identical performances. This is because both procedures

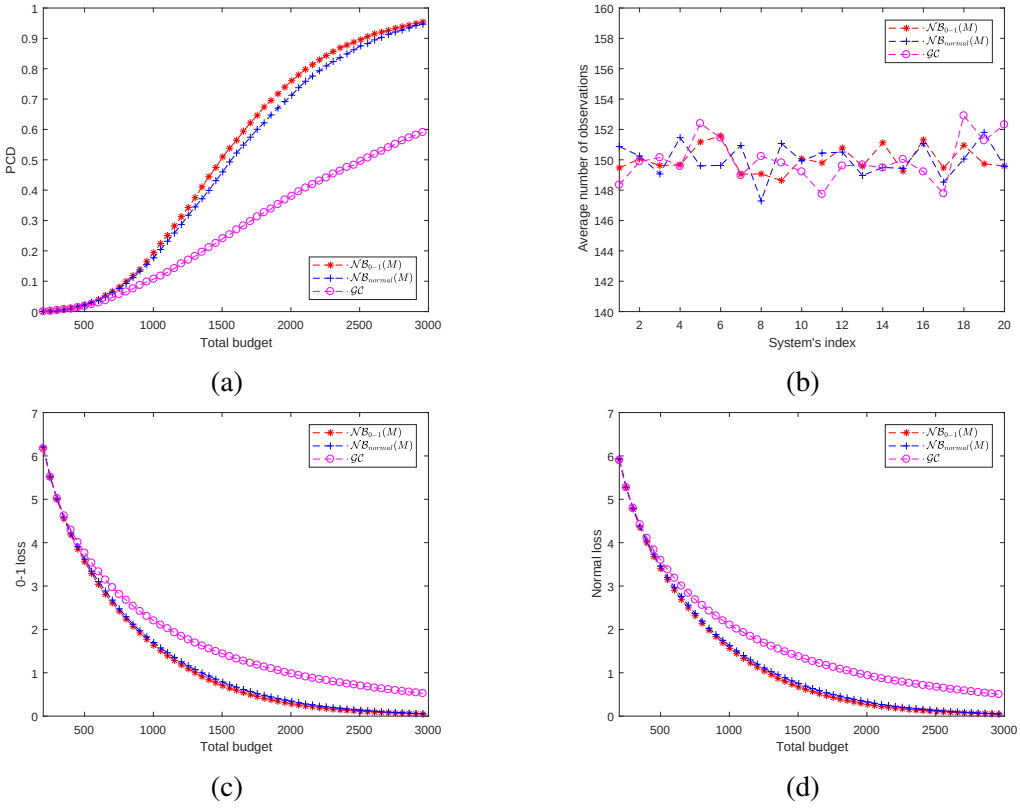


Figure 3.1: Performance comparisons among $\mathcal{NB}_{0-1}$(M), $\mathcal{NB}_{normal}$(M) and $\mathcal{GC}$ procedures under the slippage configuration. (a) PCD against different total budgets. (b) Average number of observations taken from each system after a total budget of 3000 is spent. (c) Expected 0-1 loss against different total budgets. (d) Expected normal loss against different total budgets.

select the system to take observations that minimizes the expected loss at each stage, and such a selection is not affected much by the choice of either 0-1 loss or normal loss.

3.5 Conclusion

In this chapter, we presented both two-stage and sequential Bayesian procedures that are simple to compute. Our central idea is that computing budget should be allocated so that the expected loss is minimized. To do so, we first consider that there are only two stages for taking observations, and develop budget allocation rules to minimize the expected loss from taking the second-stage observations. Then, the two-stage procedures are extended to sequential ones by iteratively minimizing the expected loss from the next stage until the to-

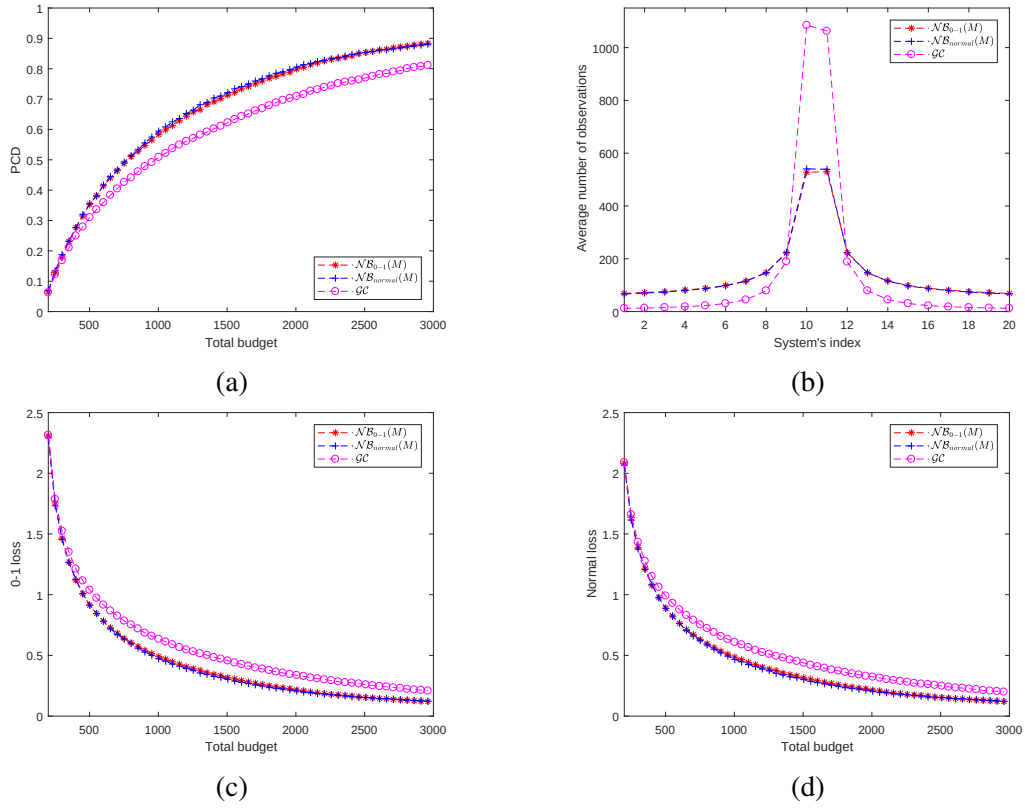


Figure 3.2: Performance comparisons among $\mathcal{NB}_{0-1}$(M), $\mathcal{NB}_{normal}$(M) and $\mathcal{GC}$ procedures under the constant difference configuration. (a) PCD against different total budgets. (b) Average number of observations taken from each system after a total budget of 3000 is spent. (c) Expected 0-1 loss against different total budgets. (d) Expected normal loss against different total budgets.

tal budget is spent. In reality, a total budget constraint arises naturally and a decision maker would usually prefer spending all the budget in multiple stages. Therefore, we propose the $\mathcal{NB}_{0-1}(M)$ and $NB_{normal}(M)$ to be used in practice. From our experiments, we see that the procedures $\mathcal{NB}_{0-1}(M)$ and $NB_{normal}(M)$ outperform the $\mathcal{GC}$ procedure throughout different configurations. The performances of these new procedures seem promising and further investigation on their performances on a broader range of applications should be expected.

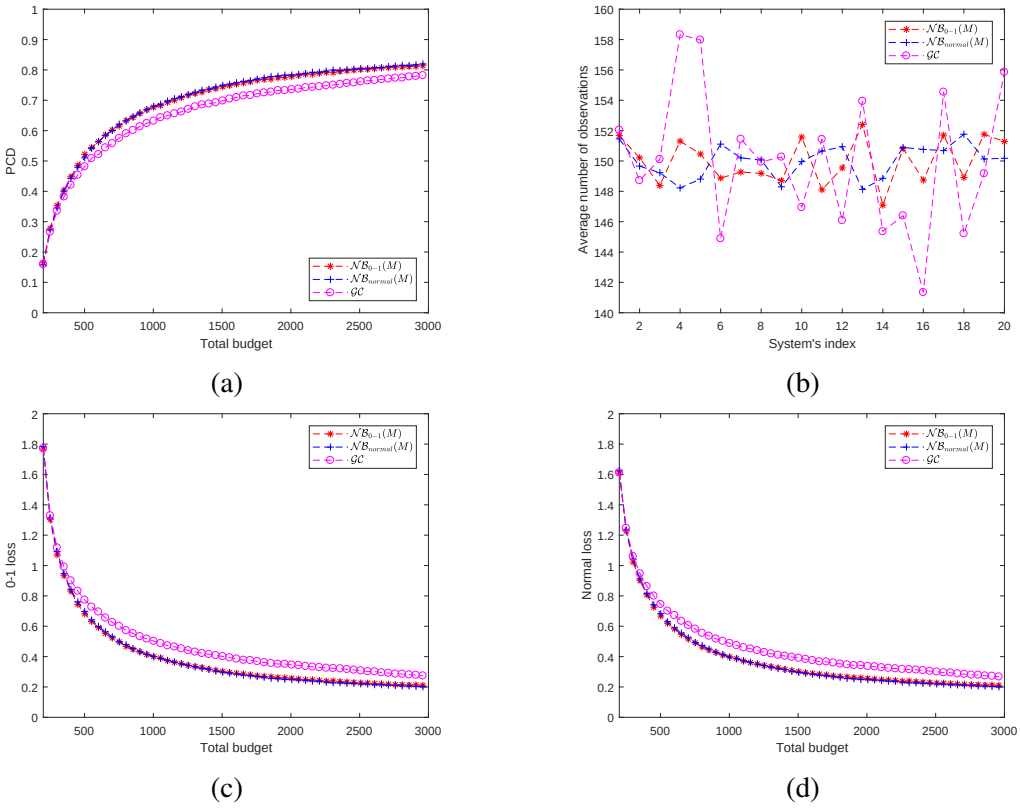


Figure 3.3: Performance comparisons among $\mathcal{NB}_{0-1}$(M), $\mathcal{NB}_{normal}$(M) and $\mathcal{GC}$ procedures under normal random configuration. (a) PCD against different total budgets. (b) Average number of observations taken from each system after a total budget of 3000 is spent. (c) Expected 0-1 loss against different total budgets. (d) Expected normal loss against different total budgets.

CHAPTER 4

FEASIBILITY DETERMINATION VIA GAUSSIAN PROCESS

So far we have discussed the problem of feasibility determination in which the simulated alternatives are considered as discrete and have no relationship each other. However, if two alternatives are similar, their performance measures are likely to be similar as well. On the other hand, if two alternatives have quite different configurations, we expect that their performance measures tend to be showing little or no relationship. Such beliefs on performance measures can be exploited to achieve better results in solving the feasibility determination problem. In this chapter, we consider the feasibility determination problem in which the alternatives are inputs of parameters that are discrete and show similar performance measures when the inputs of parameters are similar. Instead of finding a finite subset of feasible alternatives as in the previous chapters, we want to identify feasible ranges such that any alternative within those ranges is feasible. To achieve our goal, we employ a Gaussian process (GP) in modeling the performance measure of interest over the entire space of simulated alternatives. More specifically, we use a GP as a prior for the performance measures over the space of simulated alternatives, take observations from an alternative at each stage, and update the GP. We formulate the feasibility problem as an optimization problem where the objective is to maximize the expected reward. The expected reward is maximized when the feasibility decisions are correct for all alternatives. When the goal is to find an alternative with the maximum or minimum performance measure, optimization based on a GP requires an acquisition function to make sequential decisions on which alternative to sample next at each stage. A typical acquisition function calculates how much improvement in the maximum or minimum of the posterior GP is expected when an alternative is to be sampled and an alternative with the maximum acquisition function value is chosen to sample next. However, since our goal is to classify each alternative between

feasible and infeasible regions, it is desirable to sample an alternative which is expected to bring the most improvement on correct feasibility decisions over the entire space of alternatives. We develop a novel acquisition function for feasibility determination via GP. Then, we propose a Bayesian procedure that utilizes the acquisition function to solve the feasibility determination problem.

This chapter is organized as follows: in Section 4.1, we introduce necessary background information and formulate the problem; in Section 4.2, we present our acquisition function and procedure and discuss the asymptotic convergence of the proposed procedure. Section 4.3 gives details on how to evaluate the acquisition function which is based on the value of information approach for the 0-1 reward function; in Section 4.4, we address several issues in implementing the proposed procedure in practice; in Section 4.5, we present the experimental results that show the advantage of the proposed procedure followed by concluding remarks in Section 4.6.

4.1 Background

In this section, we provide necessary notations and assumptions in order to properly formulate our problem.

4.1.1 Assumptions

Let $\mathscr{X} \subset \mathbb{R}^m$ be the space of input parameters. Each alternative is specified by a vector $\mathbf{x} \in \mathscr{X}$. The mean performance measure of interest, denoted as $\mu(\mathbf{x})$, is a function of $\mathbf{x}$. Using a Bayesian formulation, we place a m-dimensional normal prior distribution to describe uncertainty about the unknown value of $\mu(\mathbf{x})$ and the collection of all $\mu(\mathbf{x})$'s, denoted as μ, follows a Gaussian process. According to [38], a Gaussian process is completely specified by its mean function and covariance function. Specifically, we have the following assumption:

Assumption 5. *For any* $\mathbf{x} \in \mathscr{X}$, $\mu(\mathbf{x}) \sim N\left(\delta(\mathbf{x}), \Sigma(\mathbf{x}, \mathbf{x})\right)$ *and the collection* $\mu \sim GP(\delta, \Sigma)$*, where* $GP(\delta, \Sigma)$ *is a Gaussian process with mean function* δ *and covariance function* Σ *defined as*

$$\begin{aligned}
\delta(\mathbf{x}) &= \mathbb{E}[\mu(\mathbf{x})], \\
\Sigma(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[\left(\mu(\mathbf{x}) - \delta(\mathbf{x})\right)\left(\mu(\mathbf{x}') - \delta(\mathbf{x}')\right)\right],
\end{aligned}$$

for any $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$.

Furthermore, consider two sets of alternatives:

$$\begin{aligned}
\mathbf{A} &= \{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(p)}\}, \\
\mathbf{B} &= \{\mathbf{x}'_{(1)}, \mathbf{x}'_{(2)}, \ldots, \mathbf{x}'_{(q)}\}.
\end{aligned}$$

Then we define

$$\vec{\delta}(\mathbf{A}) = \begin{bmatrix} \delta(\mathbf{x}_{(1)}) \\ \delta(\mathbf{x}_{(2)}) \\ \vdots \\ \delta(\mathbf{x}_{(p)}) \end{bmatrix} \text{ and } \vec{\Sigma}(\mathbf{A}, \mathbf{B}) = \begin{bmatrix} \Sigma(\mathbf{x}_{(1)}, \mathbf{x}'_{(1)}) & \Sigma(\mathbf{x}_{(1)}, \mathbf{x}'_{(2)}) & \cdots & \Sigma(\mathbf{x}_{(1)}, \mathbf{x}'_{(q)}) \\ \Sigma(\mathbf{x}_{(2)}, \mathbf{x}'_{(1)}) & \Sigma(\mathbf{x}_{(2)}, \mathbf{x}'_{(2)}) & \cdots & \Sigma(\mathbf{x}_{(2)}, \mathbf{x}'_{(q)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma(\mathbf{x}_{(p)}, \mathbf{x}'_{(1)}) & \Sigma(\mathbf{x}_{(p)}, \mathbf{x}'_{(2)}) & \cdots & \Sigma(\mathbf{x}_{(p)}, \mathbf{x}'_{(q)}) \end{bmatrix}.$$

Throughout the chapter, we abuse notation by using $\Sigma(\mathbf{x})$ to denote $\Sigma(\mathbf{x}, \mathbf{x})$ to simplify notation. Note that δ and Σ return real-valued constants while $\vec{\delta}$ and $\vec{\Sigma}$ return matrices. In practice, the functions δ and Σ are unknown. Initially, we place the prior as $\delta = \delta_0$ and $\Sigma = \Sigma_0$. A parametric family can be used to specify δ_0 and Σ_0 in terms of functions that take $\mathbf{x}$'s and a few additional parameters — known as hyper parameters — as arguments. These parameters that specify δ_0 and Σ_0 need to be estimated in general, but we will initially assume they are fully known for simplicity and discuss how to estimate their values in Section 4.4.1. After n stages of sampling, we update these functions as $\delta = \delta_n$ and

$\Sigma = \Sigma_n$, which is shown in Section 4.3.1.

In stochastic simulation, we need to simulate observations of alternative $\mathbf{x}$ to estimate its mean performance. When simulating from $\mathbf{x}$, we obtain a noisy observation $Y(\mathbf{x})$. Note that a decision maker may choose to simulate multiple observations at $\mathbf{x}$, especially when the variability of noisy observations is high. In that case, $Y(\mathbf{x})$ represents the sample mean of those multiple simulated observations. Whether $Y(\mathbf{x})$ is a single observation or a sample mean of multiple observations, it is assumed to satisfy the following assumption:

Assumption 6. *For any* $\mathbf{x} \in \mathscr{X}$, $Y(\mathbf{x}) = \mu(\mathbf{x}) + \epsilon(\mathbf{x})$ *where* $\epsilon(\mathbf{x})$ *is independent normal noise with mean zero and variance* $\sigma_s^2(\mathbf{x}) \in [0, \infty)$. *Thus,* $Y(\mathbf{x})|\mu(\mathbf{x}) \overset{iid}{\sim} \mathcal{N}\left(\mu(\mathbf{x}), \sigma_s^2(\mathbf{x})\right)$ *and* $Y(\mathbf{x}) \overset{iid}{\sim} \mathcal{N}\left(\delta(\mathbf{x}), \Sigma(\mathbf{x}) + \sigma_s^2(\mathbf{x})\right)$. *For* $\mathbf{x} \neq \mathbf{x}'$, $Y(\mathbf{x})|\mu(\mathbf{x})$ *and* $Y(\mathbf{x}')|\mu(\mathbf{x}')$ *are independent.*

The assumption that $Y(\mathbf{x})|\mu(\mathbf{x})$ and $Y(\mathbf{x}')|\mu(\mathbf{x}')$ are independent implies no common random numbers used in simulation. The sampling variance $\sigma_s^2(\mathbf{x})$ can be different for different $\mathbf{x}$ and unknown in practice, but for now we assume that they are known. We discuss how to deal with unknown $\sigma_s^2(\mathbf{x})$ later in Section 4.4.2.

4.1.2 Problem Formulation

We consider an input space $\mathscr{X} \subset \mathbb{R}^m$ and there are k alternatives on the grid of $\mathscr{X}$ that are available for simulation. Let $\mathbf{X}$ be the collection of all k alternatives. Given maximum stage T, our goal is identify a subset of $\mathbf{X}$, denoted as $\mathbf{F}$ such that $\mu(\mathbf{x}) \leq d$ for all $\mathbf{x} \in \mathbf{F}$.

We start by setting the priors $\delta = \delta_0$ and $\Sigma = \Sigma_0$. Similar as in Chapter 2, a reward function R is chosen by the decision maker to assign a reward when an alternative's feasibility is determined. The reward function is defined as follows:

$$R(\mathbf{F}; \mu, d) = \sum_{\mathbf{x} \in \mathbf{F}} r_0\left(\mu(\mathbf{x}), d\right) + \sum_{\mathbf{x} \notin \mathbf{F}} r_1\left(\mu(\mathbf{x}), d\right), \tag{4.1}$$

where r_0 and r_1 are known real-valued functions. In this chapter, we consider the 0-1

reward function only. At each stage n, we select one input $\mathbf{x}_n \in \mathbf{X}$ to sample and the set $\mathbf{F}_n$ is chosen to maximize the expected reward function. More specifically, for all $n \geq 0$,

$$\begin{aligned}\mathbf{F}_n &= \arg\max_{\mathbf{F}\subset\mathbf{X}} \mathbb{E}_n\left[R(\mathbf{F};\mu,d)\right] \\ &= \arg\max_{\mathbf{F}\subset\mathbf{X}} \left\{\sum_{\mathbf{x}\in\mathbf{F}} \mathbb{E}_n\left[r_0\left(\mu(\mathbf{x}),d\right)\right] + \sum_{\mathbf{x}\notin\mathbf{F}} \mathbb{E}_n\left[r_1\left(\mu(\mathbf{x}),d\right)\right]\right\},\end{aligned}$$

where $\mathbb{E}_n[\cdot]$ denotes the conditional expectation given the information of all observations up to stage n. A policy π is defined as a decision rule for choosing a sequence of alternatives to be sampled until the maximum stage T is reached. In particular, any policy π can be represented by an ordered sequence $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$. The optimization problem that we want to solve is

$$\sup_{\pi} \mathbb{E}^{\pi}\left[R\left(\mathbf{F};\mu,d\right)\right], \tag{4.2}$$

where $\mathbb{E}^{\pi}[\cdot]$ denotes the unconditional expectation under policy π. In the next section, we discuss our approach to solve (4.2).

4.2 New Gaussian-Based Sequential Procedure

A GP-based sequential procedure uses an acquisition function to determine which alternative to sample next. In this section, we present a novel acquisition function appropriate for feasibility determination and give the full description of our new GP-based sequential procedure. Then the asymptotic convergence of our proposed procedure is discussed.

4.2.1 Acquisition Function and Procedure

Our approach uses the value of information as our acquisition function, where the value of information at an alternative is defined as the expected increment in the expected reward function when the alternative is to be sampled next. For more description about value of information, one can refer to [71].

For a feasibility determination problem, we define the value of information as follows:

$$\text{VOI}_n(\mathbf{x}) = \mathbb{E}_n\left[\mathbb{E}_{n+1}\left[R(\mathbf{F}_{n+1};\mu,d)\right]|\mathbf{x}_{n+1}=\mathbf{x}\right] - \mathbb{E}_n\left[R(\mathbf{F}_n;\mu_n,d)\right]. \tag{4.3}$$

The analytical expression of VOI_n is presented in Section 4.3.2. At stage n, an alternative to sample next is

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbf{X}} \text{VOI}_n(\mathbf{x}).$$

Let $\mathbf{x}_n$ be the alternative sampled at stage n, and y_n be the realized observation at $\mathbf{x}_n$. Define $\mathcal{X}_n = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and $\mathcal{Y}_n = \{y_1, y_2, \ldots, y_n\}$, which are collections of sampled alternatives and observations up to stage n, respectively. By [38], the posterior mean and covariance matrices over the entire set of alternatives, $\mathbf{X}$, are updated as follows:

$$\vec{\delta}_n(\mathbf{X}) = \vec{\Sigma}_0(\mathbf{X},\mathcal{X}_n)\Lambda_n^{-1}\left[\mathcal{Y}_n - \vec{\delta}_0(\mathcal{X}_n)\right] + \vec{\delta}_0(\mathbf{X}), \tag{4.4}$$

$$\vec{\Sigma}_n(\mathbf{X},\mathbf{X}) = \vec{\Sigma}_0(\mathbf{X},\mathbf{X}) - \vec{\Sigma}_0(\mathbf{X},\mathcal{X}_n)\Lambda_n^{-1}\vec{\Sigma}_0(\mathcal{X}_n,\mathbf{X}), \tag{4.5}$$

where $\Lambda_n = \vec{\Sigma}_0(\mathcal{X}_n,\mathcal{X}_n) + D_n$ and D_n is a $n \times n$ diagonal matrix whose element on the i_{th} row and i_{th} column is $\sigma_s^2(\mathbf{x}_i)$ for $i = 1, 2, \ldots, n$. The GP-based feasibility determination ($\mathcal{GPFD}$) procedure is given in Algorithm 4.

Algorithm 4 Procedure $\mathcal{GPFD}$

1: **Setup**: Specify the set of inputs $\mathbf{X}$, threshold d and maximum of stage T.
2: **Initialization**: Set $n = 0$ and specify δ_0 and Σ_0. Choose an alternative to be sampled in the next stage as $\mathbf{x}_1 = \arg\max_{\mathbf{x}\in\mathbf{X}} \text{VOI}_0(\mathbf{x})$.
3: **Update**: Take an observation (a single observation or sample mean of multiple observations) from $\mathbf{x}_{n+1}$. Compute posterior parameters δ_{n+1} and Σ_{n+1}. Choose the alternative to be sampled in the next stage as $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbf{X}} \text{VOI}_n(\mathbf{x})$. Set $n = n+1$.
4: **Stopping Rule**: If $n < T$, go back to **Update**. Otherwise, stop the procedure and go to **Feasibility Check**.
5: **Feasibility Check**: Return the feasible set as $\mathbf{F} = \{\mathbf{x} \in \mathbf{X} : \delta_n(\mathbf{x}) \leq d\}$.

4.2.2 Convergence Property

We present the main theorem of this chapter.

Theorem 2. *As stage $n \to \infty$, $\mathbb{E}_n\left[R(\mathbf{F}_n; \mu, d)\right]$ converges to its maximum almost surely under the $\mathcal{GPFD}$ procedure with 0-1 reward function.*

Theorem 2 implies that the $\mathcal{GPFD}$ procedure samples all alternatives infinitely often as stage goes to ∞, and consequently, the expected reward converges to its maximum. The proof of Theorem 2 can be found in Appendix B.

4.3 Value of Information

In this section, we first derive the predictive distribution of posterior means and then discusses how to evaluate the value of information (i.e, VOI_n).

4.3.1 Predictive Distribution for Posterior Means

To calculate VOI_n, a predictive distribution for posterior means is needed. It is shown in [23] that (4.4) and (4.5) for matrices $\vec{\delta}_n$ and $\vec{\Sigma}_n$ can be rewritten in recursive forms of individual δ_n and Σ_n:

$$\delta_n(\mathbf{x}) = \Sigma_{n-1}(\mathbf{x}, \mathbf{x}_n)\left[\Sigma_{n-1}(\mathbf{x}_n) + \sigma_s^2(\mathbf{x}_n)\right]^{-1}\left[y_n - \delta_{n-1}(\mathbf{x}_n)\right] + \delta_{n-1}(\mathbf{x}), \tag{4.6}$$

$$\Sigma_n(\mathbf{x}, \mathbf{x}') = \Sigma_{n-1}(\mathbf{x}, \mathbf{x}') - \Sigma_{n-1}(\mathbf{x}, \mathbf{x}_n) \cdot \Sigma_{n-1}(\mathbf{x}', \mathbf{x}_n) / \left[\Sigma_{n-1}(\mathbf{x}_n) + \sigma_s^2(\mathbf{x}_n)\right], \tag{4.7}$$

for any $\mathbf{x}, \mathbf{x}' \in \mathbf{X}$.

Suppose at stage $n+1$, the alternative to be sampled is $\mathbf{x}_{n+1}$ and Y_{n+1} is the variable of the corresponding observation. Note that the observation is unknown at stage n and therefore, we denote it as a random variable Y_{n+1}. Under Assumption 6, we have that

$$Y_{n+1} \sim \mathcal{N}\left(\delta_n(\mathbf{x}_{n+1}), \Sigma_n(\mathbf{x}_{n+1}) + \sigma_s^2(\mathbf{x}_{n+1})\right). \tag{4.8}$$

Following (4.6),

$$\delta_{n+1}(\mathbf{x}) = \Sigma_n(\mathbf{x}, \mathbf{x}_{n+1}) \left[\Sigma_n(\mathbf{x}_{n+1}) + \sigma_s^2(\mathbf{x}_{n+1})\right]^{-1} \left[Y_{n+1} - \delta_n(\mathbf{x}_{n+1})\right] + \delta_n(\mathbf{x}) \qquad (4.9)$$

for any $\mathbf{x} \in \mathbf{X}$.

Since $\delta_{n+1}(\mathbf{x})$ is a linear function of Y_{n+1}, the predictive distribution of $\delta_{n+1}(\mathbf{x})$ given $\mathbf{x}_{n+1}$ is a normal distribution as follows:

$$\delta_{n+1}(\mathbf{x}) \sim \mathcal{N}\left(\delta_n(\mathbf{x}), \Sigma_n^2(\mathbf{x}, \mathbf{x}_{n+1}) / \left[\Sigma_n(\mathbf{x}_{n+1}) + \sigma_s^2(\mathbf{x}_{n+1})\right]\right),$$

for any $\mathbf{x} \in \mathbf{X}$.

4.3.2 Evaluation of the Value of Information

Now we present the analytical expression of the value of information for the 0-1 reward function. First, let

$$r(\mathbf{F}, \mathbf{x}; \mu, d) = r_0\left(\mu(\mathbf{x}), d\right) \cdot \mathrm{I}\left(\mathbf{x} \in \mathbf{F}\right) + r_1\left(\mu(\mathbf{x}), d\right) \cdot \mathrm{I}\left(\mathbf{x} \notin \mathbf{F}\right).$$

From the definition of the reward function in (4.1), we have

$$\begin{aligned} R(\mathbf{F}; \mu, d) &= \sum_{\mathbf{x} \in \mathbf{X}} \left[r_0\left(\mu(\mathbf{x}), d\right) \cdot \mathrm{I}\left(\mathbf{x} \in \mathbf{F}\right) + r_1\left(\mu(\mathbf{x}), d\right) \cdot \mathrm{I}\left(\mathbf{x} \notin \mathbf{F}\right)\right] \\ &= \sum_{\mathbf{x} \in \mathbf{X}} r\left(\mathbf{F}, \mathbf{x}; \mu, d\right), \end{aligned}$$

where $\mathrm{I}(\cdot)$ is the indicator function. Then, based on the definition of $\mathrm{VOI}_n(\mathbf{x})$ in (4.3), we have

$$\mathrm{VOI}_n(\mathbf{x}) = \sum_{\mathbf{x}' \in X} \mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d) | \mathbf{x}_{n+1} = \mathbf{x}\right] - \sum_{\mathbf{x}' \in X} \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right]. \qquad (4.10)$$

As in Chapter 2, the 0-1 reward function is defined as $r_0(\mu(\mathbf{x}), d) = \mathrm{I}(\mu(\mathbf{x}) \leq d)$ and $r_1(\mu(\mathbf{x}), d) = \mathrm{I}(\mu(\mathbf{x}) > d)$. Consequently, $\mathbb{E}_n[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)] = \Phi\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right)$ for any $\mathbf{x}' \in \mathbf{X}$. Here, $\Phi(\cdot)$ is the cumulative distribution function of a standard normal random variable. The rest of this section is on the calculation of $\mathbb{E}_n[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}]$.

Suppose at stage $n+1$, we choose $\mathbf{x}_{n+1} = \mathbf{x}$ to sample. For any $\mathbf{x}' \in \mathbf{X}$, $\delta_{n+1}(\mathbf{x}')$ are given in (4.9), which is a function of Y_{n+1}. Define $h(Y_{n+1}) = \delta_{n+1}(\mathbf{x}')$, and we have

$$\mathbb{E}_{n+1}[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}, Y_{n+1}] = \begin{cases} \Phi\left(\dfrac{d - h(Y_{n+1})}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right), & \text{if } h(Y_{n+1}) \leq d, \\ 1 - \Phi\left(\dfrac{d - h(Y_{n+1})}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right), & \text{otherwise.} \end{cases} \quad (4.11)\text{–}(4.12)$$

Next, we discuss how to compute (4.11) and (4.12) in different cases.

Case 1: $\Sigma_n(\mathbf{x}', \mathbf{x}) > 0$

From (4.8), it is easy to see that $\frac{Y_{n+1}-\delta_n(\mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x})+\sigma_s^2(\mathbf{x})}}$ has the same distribution as the standard normal random variable Z. Then (4.9) can be re-written as

$$\delta_{n+1}(\mathbf{x}') = \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot Z + \delta_n(\mathbf{x}')$$

and with the definition $h(Y_{n+1}) = \delta_{n+1}(\mathbf{x}')$, the event $\{h(Y_{n+1}) \leq d\}$ is equivalent to

$$\left\{ Z \leq \frac{[d - \delta_n(\mathbf{x}')]\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}}{\Sigma_n(\mathbf{x}', \mathbf{x})} \right\}.$$

Let

$$\rho = \frac{[d - \delta_n(\mathbf{x}')]\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}}{\Sigma_n(\mathbf{x}', \mathbf{x})}.$$

Then the expected reward at $\mathbf{x}'$ given $\mathbf{x}_{n+1} = \mathbf{x}$ is

$$\begin{aligned}
\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] &= \mathbb{E}_n\left[\mathbb{E}_{n+1}\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}, Y_{n+1}\right]|\mathbf{x}_{n+1} = \mathbf{x}\right] \\
&= \mathbb{E}_n\left[\mathbb{E}_{n+1}\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}, Z\right]|\mathbf{x}_{n+1} = \mathbf{x}\right] \\
&= \int_{-\infty}^{\rho} \Phi\left(\frac{d - \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot z - \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right) \phi(z) dz \\
&+ \int_{\rho}^{\infty} \Phi\left(\frac{-d + \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot z + \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right) \phi(z) dz.
\end{aligned} \tag{4.13}$$

We now compute the first and second terms in (4.13) separately. First, we have

$$\text{the first term in (4.13)} = \Pr\left\{Z_1 \leq \rho, Z_2 \leq \frac{d - \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot Z_1 - \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right\}, \tag{4.14}$$

where Z_1 and Z_2 are independent standard normal random variables.

Let

$$Z_3 = Z_2 - \frac{d - \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot Z_1 - \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}.$$

Then

$$(4.14) = \Pr\left\{Z_1 \leq \rho, Z_3 \leq 0\right\}.$$

Vector (Z_1, Z_3) follows a bivariate normal distribution with mean $M_1 = \begin{bmatrix} m_1 \\ m_3 \end{bmatrix}$ and covari-

ance matrix $S_1 = \begin{bmatrix} s_{11} & s_{13} \\ s_{31} & s_{33} \end{bmatrix}$ where

$$
\begin{aligned}
m_1 =& \mathbb{E}[Z_1] = 0; \\
m_3 =& \mathbb{E}[Z_3] = \frac{\delta_n(\mathbf{x}') - d}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}; \\
s_{11} =& \text{Var}[Z_1] = 1; \\
s_{33} =& \text{Var}[Z_3] = 1 + \frac{\Sigma_n(\mathbf{x}', \mathbf{x})^2}{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}; \text{ and} \\
s_{13} =& s_{31} = \text{Cov}\left[Z_1, Z_3\right] = \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}}.
\end{aligned}
$$

Therefore, the first term in (4.13) can be computed using the cumulative distribution function of a bivariate normal vector with mean M_1 and covariance matrix S_1 at $(\rho, 0)$.

Next, we compute the second term in (4.13). Using a substitution $z' = -z$, we have

$$
\begin{aligned}
\text{the second term in (4.13)} =& \int_{-\infty}^{-\rho} \Phi\left(\frac{-d - \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot z' + \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} \right) \phi(z') dz' \\
=& \text{Pr}\left\{ Z_1' \leq -\rho, Z_2' \leq \frac{-d - \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot Z_1' + \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} \right\},
\end{aligned}
\tag{4.15}
$$

where Z_1' and Z_2' are independent standard normal random variables. Let

$$
Z_3' = Z_2' + \frac{d + \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}} \cdot Z_1' - \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}.
$$

Then

$$
(4.15) = \text{Pr}\left\{ Z_1' \leq -\rho, Z_3' \leq 0 \right\}. \tag{4.16}
$$

Similar to the previous case, (Z_1', Z_3') follows a bivariate normal distribution with mean

$M_2 = \begin{bmatrix} m_1' \\ m_3' \end{bmatrix}$ and covariance matrix $S_2 = \begin{bmatrix} s_{11}' & s_{13}' \\ s_{31}' & s_{33}' \end{bmatrix}$ where

$$\begin{aligned}
m_1' =& \mathbb{E}[Z_1'] = 0; \\
m_3' =& \mathbb{E}[Z_3'] = \frac{d - \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}; \\
s_{11}' =& \text{Var}[Z_1'] = 1; \\
s_{33}' =& \text{Var}[Z_3'] = 1 + \frac{\Sigma_n(\mathbf{x}', \mathbf{x})^2}{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}; \text{ and} \\
s_{13}' =& s_{31}' = \text{Cov}\left[Z_1', Z_3'\right] = \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}}.
\end{aligned}$$

Therefore, the second term in (4.13) can be computed using the cumulative distribution function of a bivariate normal vector with mean M_2 and covariance S_2 at $(-\rho, 0)$.

Case 2: $\Sigma_n(\mathbf{x}', \mathbf{x}) < 0$

The proof for this case can be derived in a similar way as in 4.3.2. Here, we directly present the results:

$$\mathbb{E}_n\left[r(\mathbf{F}, \mathbf{x}'; \mu, d) | \mathbf{x}_{n+1} = \mathbf{x}\right] = \text{Pr}\left\{Z_1^{"} \leq \rho, Z_3^{"} \leq 0\right\} + \text{Pr}\left\{Z_1^{"'} \leq -\rho, Z_3^{"'} \leq 0\right\}, \tag{4.17}$$

where $(Z_1^{"}, Z_3^{"})$ follows a bivariate normal distribution with the following mean and covariance matrix

$$\begin{bmatrix} 0 \\ \frac{d - \delta_n(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} \end{bmatrix} \text{ and } \begin{bmatrix} 1 & \frac{-\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}} \\ \frac{-\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}} & 1 + \frac{\Sigma_n(\mathbf{x}', \mathbf{x})^2}{\Sigma_{n+1}(\mathbf{x}') \cdot [\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})]}, \end{bmatrix}$$

and $(Z_1^{"'}, Z_3^{"'})$ follows a bivariate normal distribution with mean and covariance matrix

$$\begin{bmatrix} 0 \\ \frac{\delta_n(\mathbf{x}')-d}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} \end{bmatrix} \text{ and } \begin{bmatrix} 1 & \frac{-\Sigma_n(\mathbf{x}',\mathbf{x})}{\sqrt{\Sigma_{n+1}(\mathbf{x}')\cdot[\Sigma_n(\mathbf{x})+\sigma_s^2(\mathbf{x})]}} \\ \frac{-\Sigma_n(\mathbf{x}',\mathbf{x})}{\sqrt{\Sigma_{n+1}(\mathbf{x}')\cdot[\Sigma_n(\mathbf{x})+\sigma_s^2(\mathbf{x})]}} & 1+\frac{\Sigma_n(\mathbf{x}',\mathbf{x})^2}{\Sigma_{n+1}(\mathbf{x}')\cdot[\Sigma_n(\mathbf{x})+\sigma_s^2(\mathbf{x})]} \end{bmatrix}.$$

Case 3: $\Sigma_n(\mathbf{x}', \mathbf{x}) = 0$

Under Assumption 5, this case essentially means that $\mu(\mathbf{x}')$ and $\mu(\mathbf{x})$ have no relationship and Y_{n+1} does not affect $\mu(\mathbf{x}')$. As a result, we have that $\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] = \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right]$.

4.4 Implementation Issues

In this section, we discuss several implementation issues for the the $\mathcal{GPFD}$ procedure. More specifically, we discuss how to specify the initial prior mean and covariance matrix, δ_0 and Σ_0, how to deal with unknown and unequal sample variances across $\mathbf{x}$'s, and how to overcome heavy computation in evaluating VOI_n.

4.4.1 Estimation of Prior Mean and Covariance Functions

We use parametric forms to specify δ_0 and Σ_0. One common practice for setting δ_0 is $\delta_0(\mathbf{x}) = C$ for all $\mathbf{x} \in \mathbf{X}$, where C is a constant. At each stage n, we estimate the constant C by the mean of all samples up to stage n, which is denoted as $\bar{Y}_n$.

There are many commonly used families of covariance functions, which are introduced in [38]. In real practice, one should choose an appropriate covariance function that best fits any known property of the objective function. In our experiments, we adopt the Matérn class of covariance functions, which is shown to be very effective in many empirical analyses from [**SLA**]. The Matérn class of covariance functions is given as

$$\Sigma_0(\mathbf{x}, \mathbf{x}') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\|\mathbf{x} - \mathbf{x}'\|}{\ell} \right),$$

where σ, ν and ℓ are positive parameters, $\|\cdot\|$ is the L2 norm, $\Gamma(\cdot)$ is the gamma function, and K_ν is a modified Bessel function (see [72]). The parameter ν controls the smoothness of the estimated Gaussian process, and we set $\nu = 1.5$, which is a popular choice in machine learning according to [38]. For more discussion on the Matérn class of covariance functions, one can also refer to [73]. For parameters σ and ℓ, we estimate them by maximizing the log-marginal likelihood. Algorithms to estimate the parameters can be found in [74] and [75].

4.4.2 Estimation of Unknown Sampling Variances

Although the sampling variances $\sigma_s^2(\mathbf{x})$'s could be different for different values of $\mathbf{x}$ in reality, it requires much more efforts in estimating all $\sigma_s^2(\mathbf{x})$'s individually. Especially in Bayesian feasibility determination, one usually assumes that observations are expensive to take (in terms of money or time) and the total number of observations that the decision maker is willing to spend is relatively small. In addition, unknown sampling variances are usually updated as more observations are available by maximizing the log-marginal likelihood and it is also computationally difficult to solve the maximization for each $\sigma_s^2(\mathbf{x})$. Therefore, estimating all different $\sigma_s^2(\mathbf{x})$'s is considered to be unrealistic in the GP-based Bayesian approach. Instead, the usual practice is to assume equal sampling variances and then find it by using the maximum likelihood estimator for the equal sampling variances. There exist some works such as [76] and [77] that address heteroscedastic Gaussian Processes, but we do not incorporate their works due to the above reasons in this paper.

4.4.3 Acceleration of $\mathcal{GPFD}$

The complexity of each stage in the $\mathcal{GPFD}$ procedure is $O(k^2)$ for k number of alternatives, because the procedure computes the expected increment in reward over all alternatives for every potential candidate. As k increases, the computation of VOI_n at each stage can be quite burdensome. We can accelerate the procedure by calculating the expected in-

crements in reward only for neighboring alternatives of each potential candidate alternative. More specifically, we revise the computation of value of information as follows:

$$\tilde{\text{VOI}}_n(\mathbf{x}) = \sum_{\mathbf{x}' \in \text{NB}(\mathbf{x})} \mathbb{E}_n \left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d) | \mathbf{x}_{n+1} = \mathbf{x} \right] - \sum_{\mathbf{x}' \in \text{NB}(\mathbf{x})} \mathbb{E}_n \left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d) \right], \quad (4.18)$$

where $\text{NB}(\mathbf{x}) = \{\mathbf{x}' \in \mathbf{X} : \|\mathbf{x}' - \mathbf{x}\| \leq w\}$ for some neighborhood distance $w > 0$ chosen by the decision maker.

4.5 Experimental Results

In this section, we present the experimental results that show the advantage of our proposed procedures. First, we use a one-dimensional input example to compare the performances of the full version of $\mathcal{GPFD}$ procedure (where VOI_n is used) and the accelerated version (where $\tilde{\text{VOI}}_n$ is used), and to test the effectiveness of the accelerated version. Then, we further illustrate the performance advantage of the accelerated $\mathcal{GPFD}$ procedure using a more complex two-dimensional input example.

To measure the performance of procedure, we record the following proportion of correct decisions at each stage in each replication defined as

$$\frac{\sum_{\mathbf{x} \in \mathbb{F}} \text{I}\left(\mathbf{x} \in \mathbf{F}_\text{n}\right) + \sum_{\mathbf{x} \notin \mathbb{F}} \text{I}\left(\mathbf{x} \notin \mathbf{F}_\text{n}\right)}{k},$$

where $\mathbb{F}$ is the true set of feasible alternatives and $\mathbf{F}_n$ is the set of feasible alternatives returned by the procedure at stage n. Then we define $\text{PCD}(n)$ as the average of the above proportions over 100 replications. Also, we report the average number of observations taken from an alternative $\mathbf{x}$ when the maximum stage T is achieved, which is denoted as $\text{OBS}(\mathbf{x})$, over 100 replications.

Throughout all experiments, we compare our proposed procedures against a random search with GP procedure. At each stage, the random search with GP procedure chooses

the next alternative to sample by randomly selecting an $\mathbf{x} \in \mathbf{X}$ with equal probability $1/k$. Then it takes an observation, updates the posterior mean and covariance functions, terminates and makes final feasibility decisions in the same way as the $\mathcal{GPFD}$ procedure.

Procedures such as the $\mathcal{GC}$ procedure from [32] and those introduced in Chapter 3 also aim to solve a feasibility determination problem with a total budget for simulation. However, those procedures require an initial sample from each alternative in order to estimate its prior mean and prior variance, which are two crucial elements in deciding how to assign a sampling budget in the next stage. In addition, since those procedures rely on asymptotic approximations that are unlikely to be accurate when the total budget is small. Both reasons suggest that those procedures are not suitable for our problem, in which the total budget is much smaller than the number of alternatives. As shown later, we only consider spending a total number of observations that is much less than the number of alternatives in each experiment.

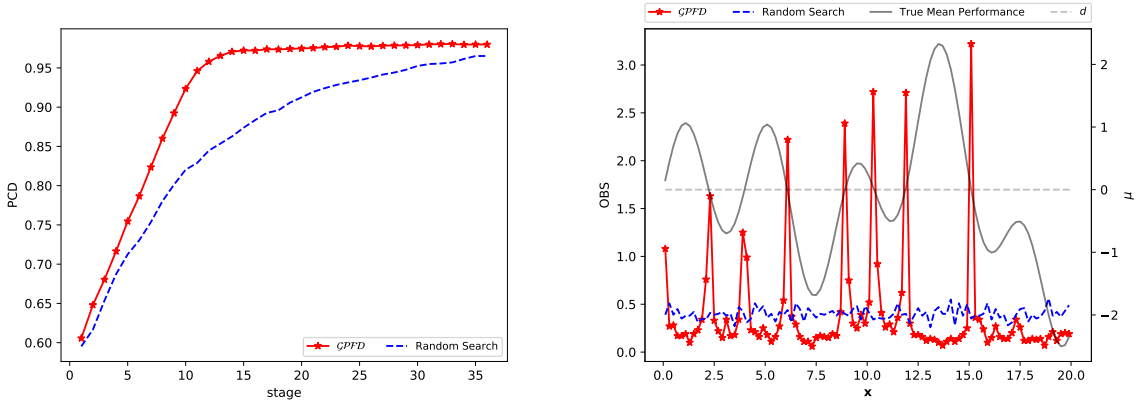
4.5.1 One-Dimensional Example

We consider $\mathbf{X} = \{0.1, 0.3, 0.5, \ldots, 19.9\}$ that contains 100 alternatives. The true mean of performance is $\mu(\mathbf{x}) = \frac{\mathbf{x}}{10} \cdot \sin(0.189\pi\mathbf{x}) + \sin(0.4789\pi\mathbf{x})$ for each $\mathbf{x} \in \mathbf{X}$. At each stage n, we take a single observation from a chosen alternative until the maximum stage $T = 36$ is reached. Our goal is to identify all alternatives whose mean performances are less than or equal to 0 (i.e., $d = 0$). To estimate the prior parameters, we randomly sample four alternatives and take one observation from each of the four randomly sampled alternatives. We conduct the experiments based on different settings of factors such as sampling variances $\sigma_s^2(\mathbf{x})$'s and neighborhood distances w's (for the accelerated procedure). For the equal sampling variances case, we set the standard deviation, $\sigma_s(\mathbf{x})$ to be constant for all $\mathbf{x} \in \mathbf{X}$. For the unequal sampling variances case, each sampling standard deviation $\sigma_s(\mathbf{x})$ is proportional to the magnitude of $|\mu(\mathbf{x}) - d|$. For the accelerated $\mathcal{GPFD}$ procedure, we test $w = 10, 6, 2$ to run the experiments.

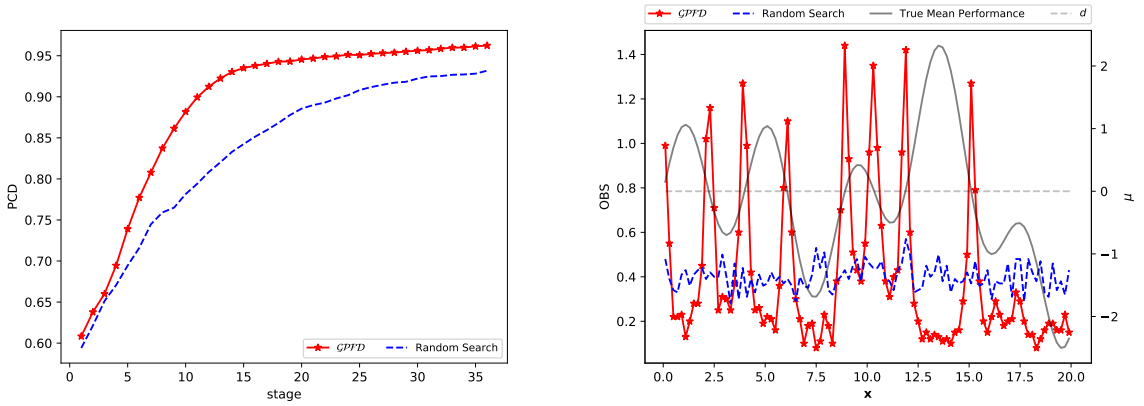
The graphs on the left side of Figure 4.1 show PCD(n) while those on the right side show OBS($\mathbf{x}$) for three different levels of $\sigma_s(\mathbf{x})$. The $\mathcal{GPFD}$ procedure achieves quite high PCD's for all three cases in just 36 stages with visiting only 40 alternatives (4 randomly sampled alternatives and 36 alternatives visited during the 36 stages) and obtaining 40 observations. The procedure also outperforms the random search with GP in all cases. From the graphs on the right side of Figure 4.1, we see that the random search with GP assigns relatively uniform numbers of observations over the 100 alternatives. On the other hand, the $\mathcal{GPFD}$ procedure assigns more observations to the alternatives whose means are close to the threshold (i.e., barely feasible/infeasible alternatives) while clearly feasible/infeasible alternatives receive fewer numbers of observations.

Figures 4.2 – 4.4 show the performances of the accelerated $\mathcal{GPFD}$ procedures under equal sampling variances. We see that as w decreases, the performance of $\mathcal{GPFD}$ procedure degrades but only slightly compared to the full version of $\mathcal{GPFD}$. The accelerated procedure still performs better than the random search with GP in all cases.

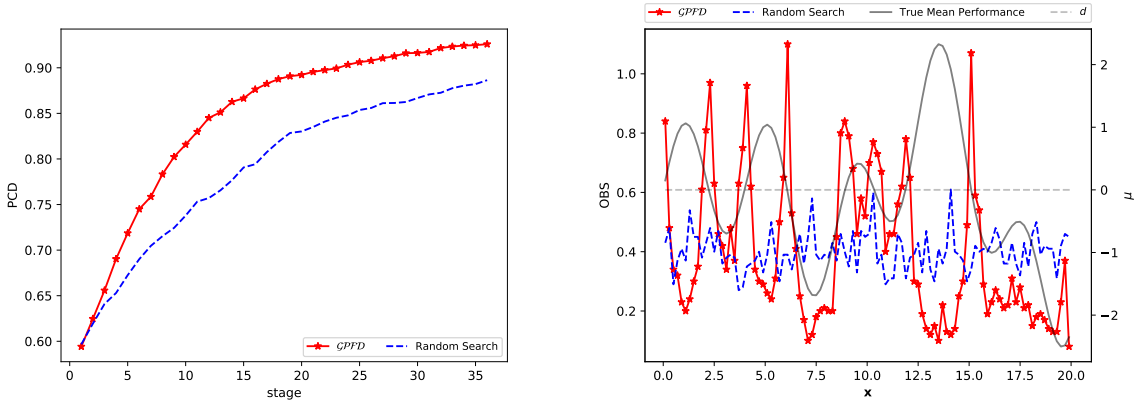
Figures 4.5 – 4.8 are counterparts of Figures 4.1 – 4.4 where sampling variances are unequal. More specifically, Figure 4.5 shows the performance of the full $\mathcal{GPFD}$ procedure and Figures 4.6 – 4.8 show the performances of the accelerated $\mathcal{GPFD}$ procedure with $w = 10, 6, 2$ under three different configurations of unequal sampling variances. We observe a similar tendency in PCD(n) and OBS($\mathbf{x}$) as in the equal sampling variances case. In fact, for each $C_1 = C_2$, we achieve higher PCD's than the equal sampling variances case because the barely feasible/infeasible alternatives have smaller sampling variances. Similar to the equal sampling variances case, barely feasible/infeasible alternatives receive more observations. However, alternatives close to those with large sampling variances (for example, $\mathbf{x} = 8.9$ and 15.1) receive even more observations.



(a) $\sigma_s(\mathbf{x}) = 0.1$

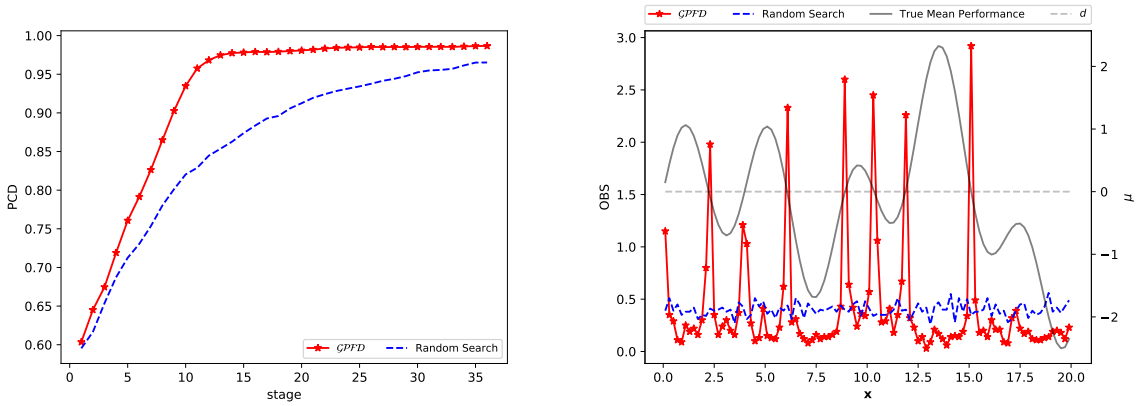


(b) $\sigma_s(\mathbf{x}) = 0.3$

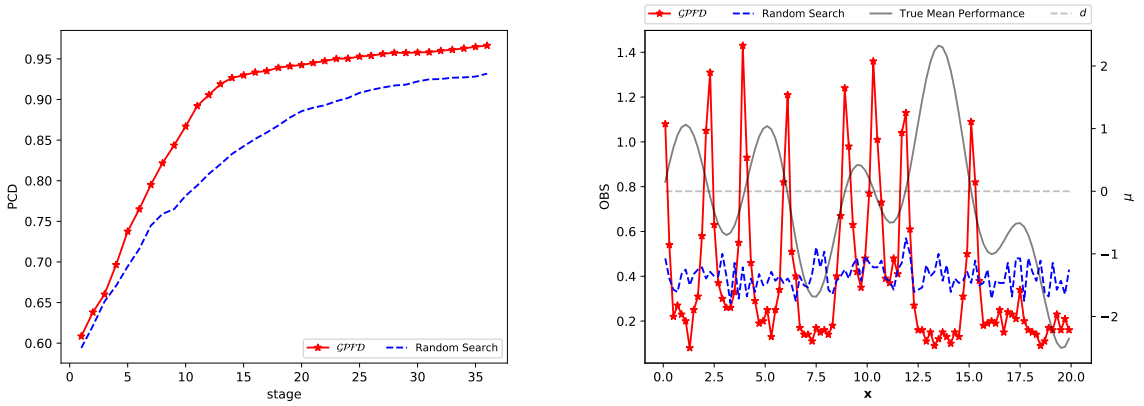


(c) $\sigma_s(\mathbf{x}) = 0.5$

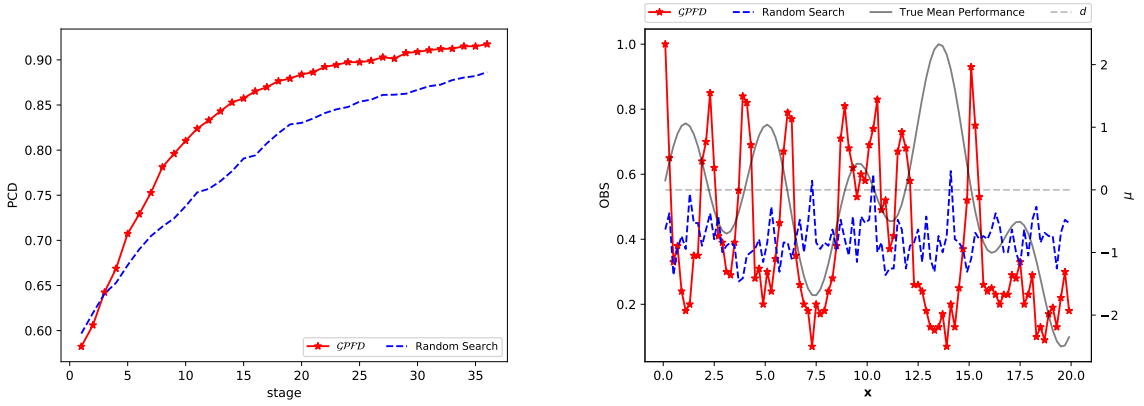
Figure 4.1: One-dimensional experimental results on equal sampling variances when the full version of $\mathcal{GPFD}$ is used.



(a) $\sigma_s(\mathbf{x}) = 0.1$

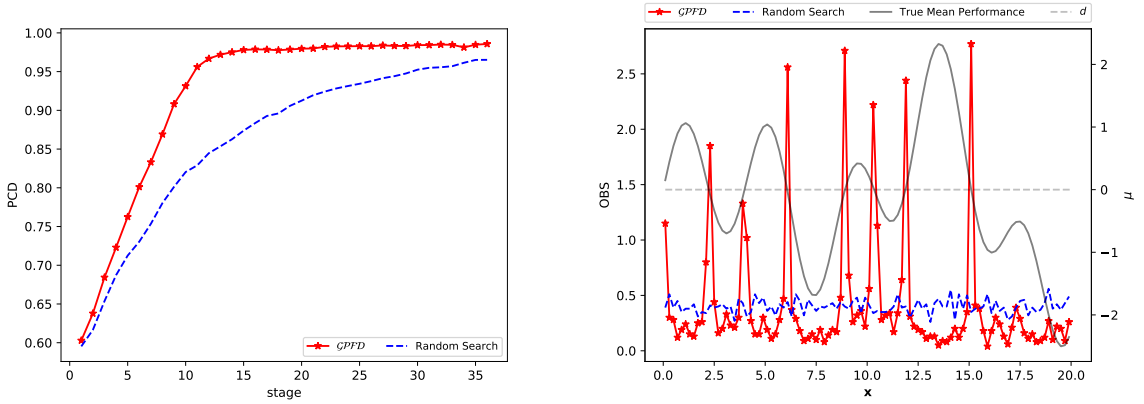


(b) $\sigma_s(\mathbf{x}) = 0.3$

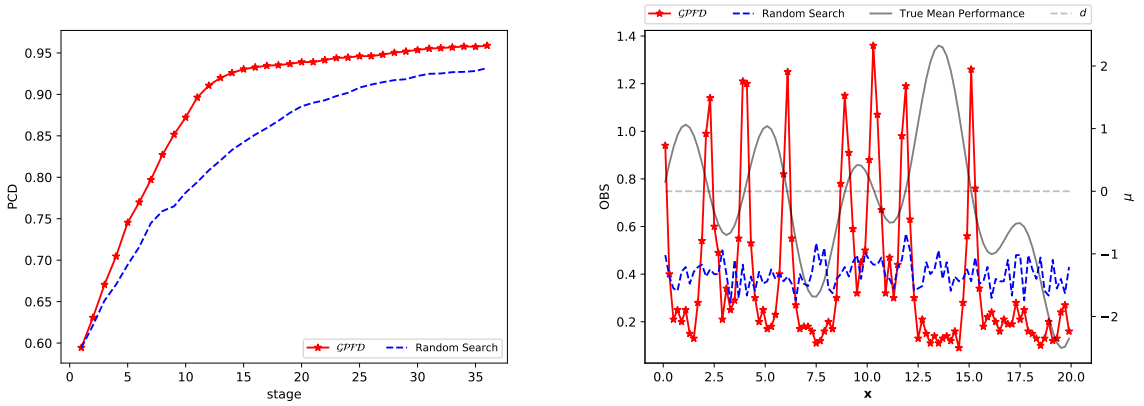


(c) $\sigma_s(\mathbf{x}) = 0.5$

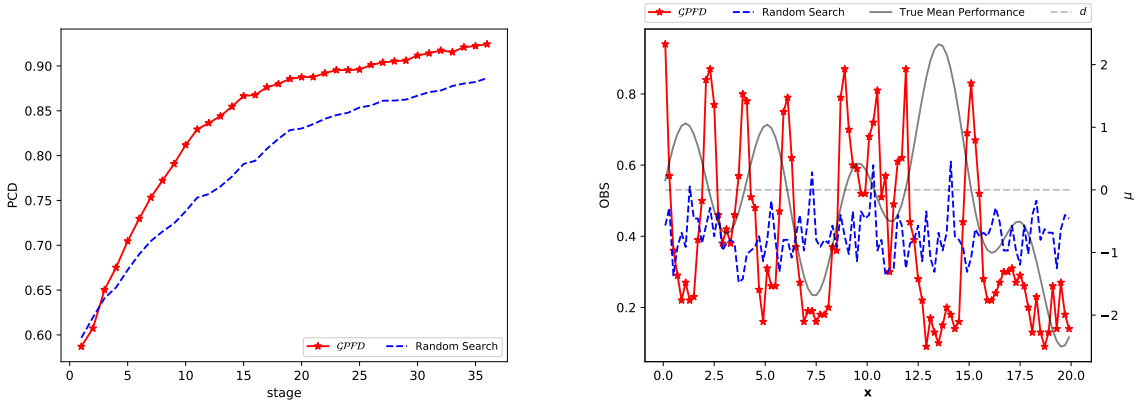
Figure 4.2: One-dimensional experimental results on equal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = 10$.



(a) $\sigma_s(\mathbf{x}) = 0.1$

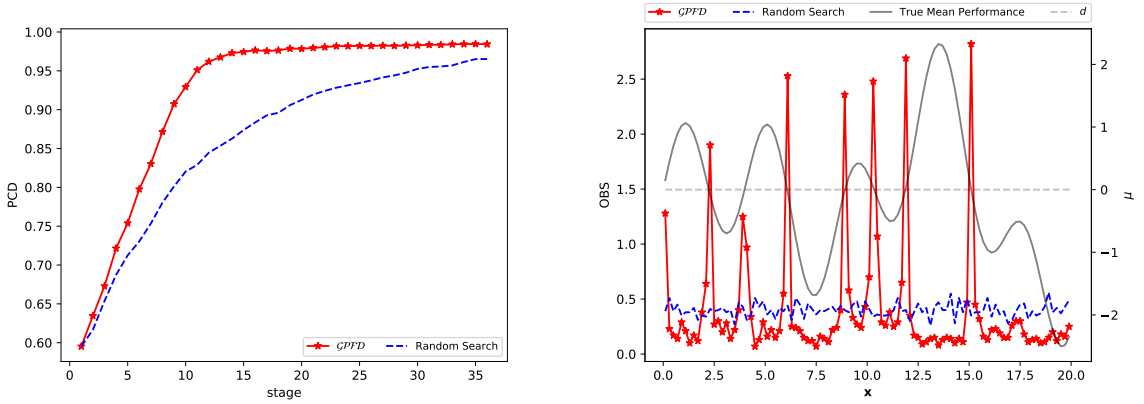


(b) $\sigma_s(\mathbf{x}) = 0.3$

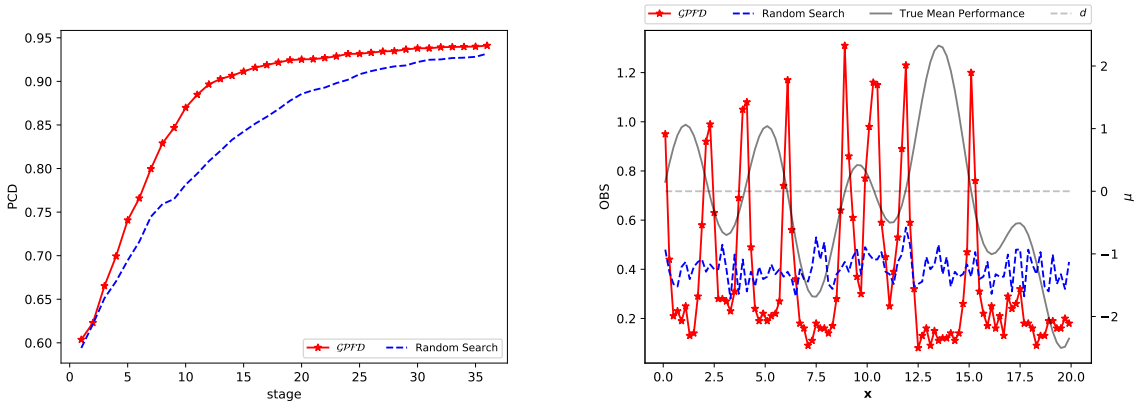


(c) $\sigma_s(\mathbf{x}) = 0.5$

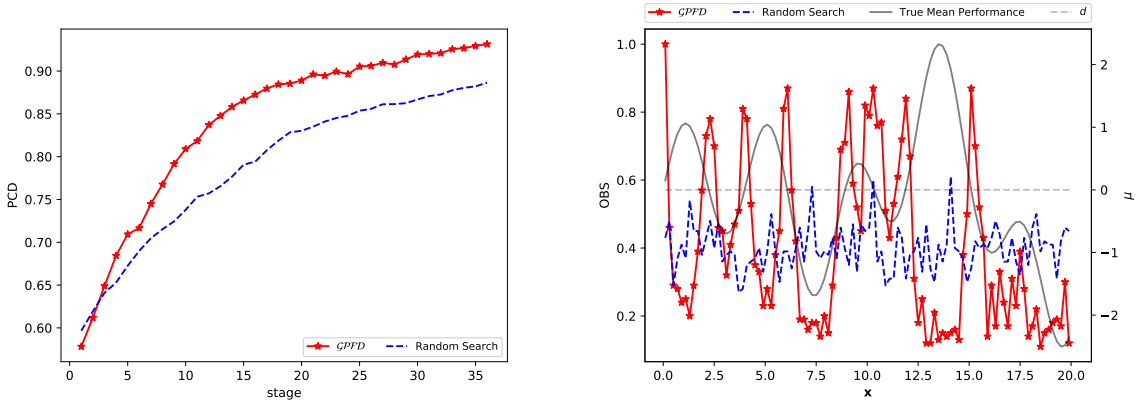
Figure 4.3: One-dimensional experimental results on equal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = 6$.



(a) $\sigma_s(\mathbf{x}) = 0.1$

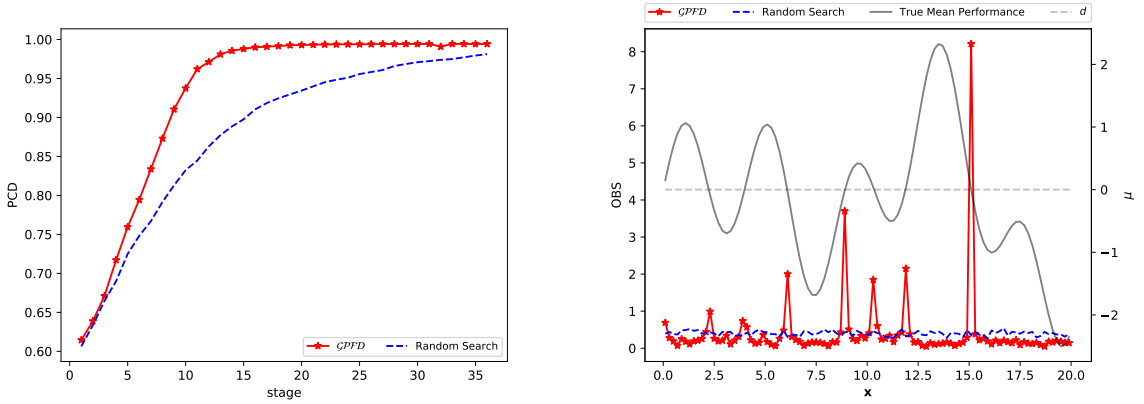


(b) $\sigma_s(\mathbf{x}) = 0.3$

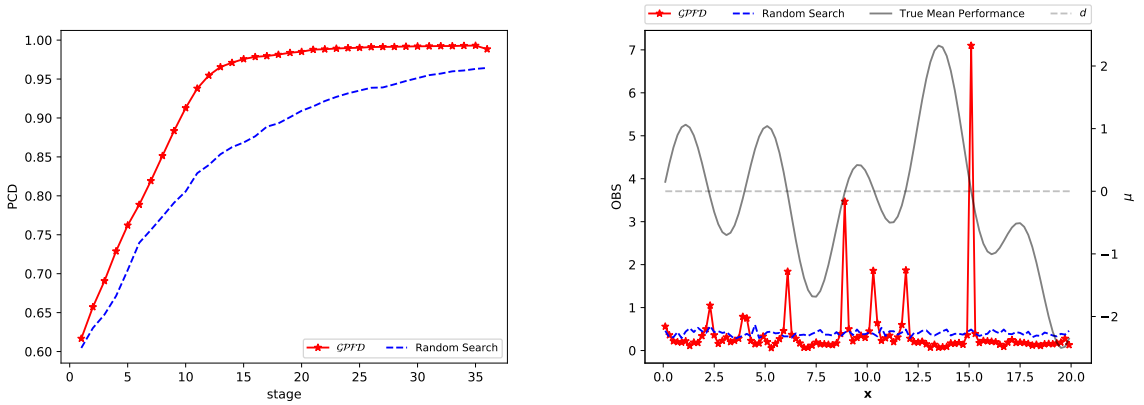


(c) $\sigma_s(\mathbf{x}) = 0.5$

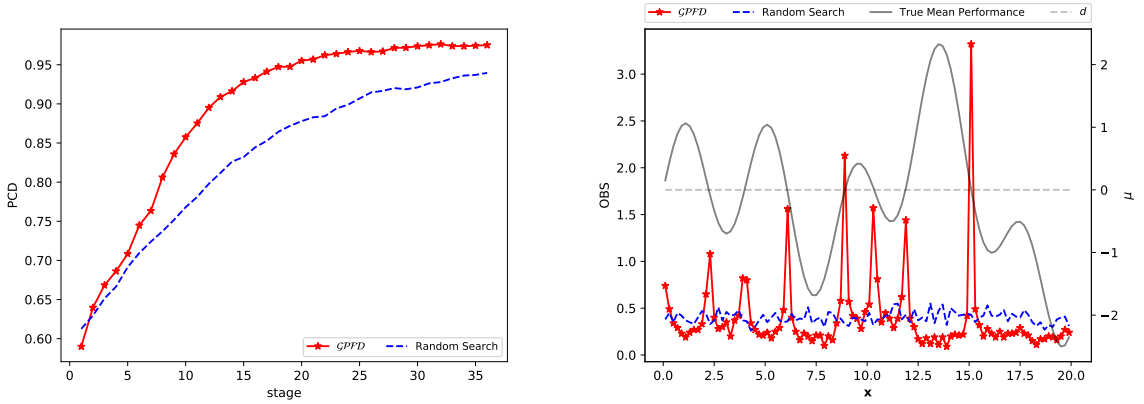
Figure 4.4: One-dimensional experimental results on equal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = 2$.



(a) $\sigma_s(\mathbf{x}) = 0.1|\mu(\mathbf{x}) - d|$

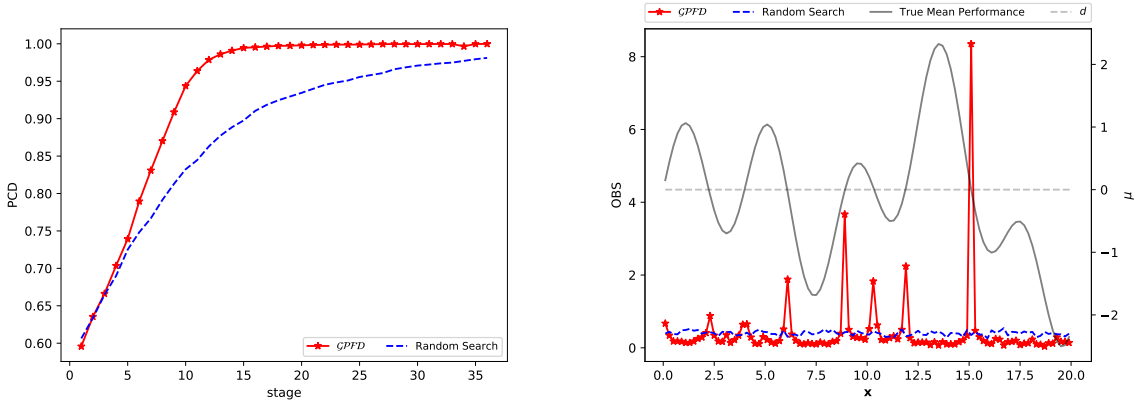


(b) $\sigma_s(\mathbf{x}) = 0.3|\mu(\mathbf{x}) - d|$

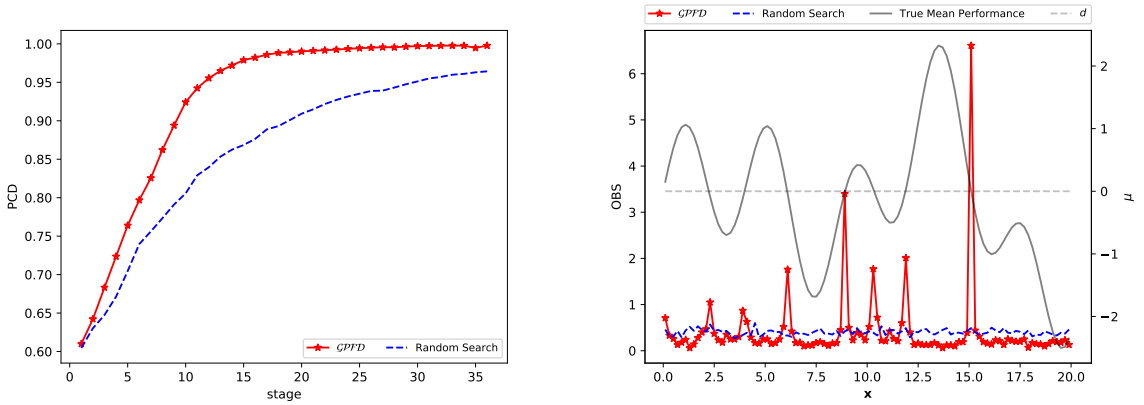


(c) $\sigma_s(\mathbf{x}) = 0.5|\mu(\mathbf{x}) - d|$

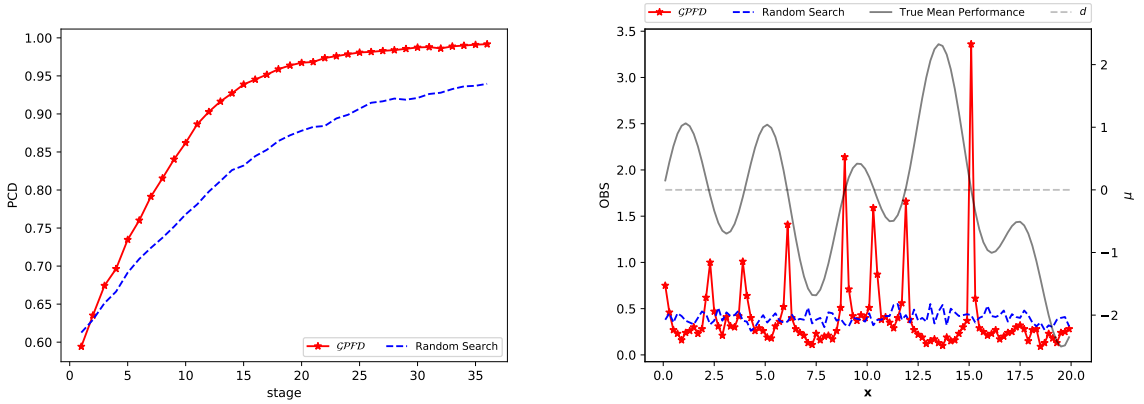
Figure 4.5: One-dimensional experimental results on unequal sampling variances when the full version of $\mathcal{GPFD}$ is used.



(a) $\sigma_s(\mathbf{x}) = 0.1|\mu(\mathbf{x}) - d|$

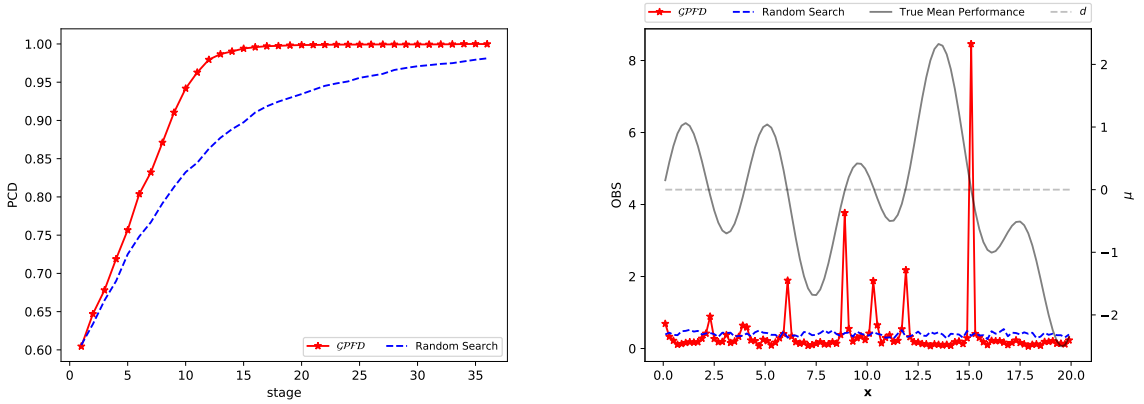


(b) $\sigma_s(\mathbf{x}) = 0.3|\mu(\mathbf{x}) - d|$

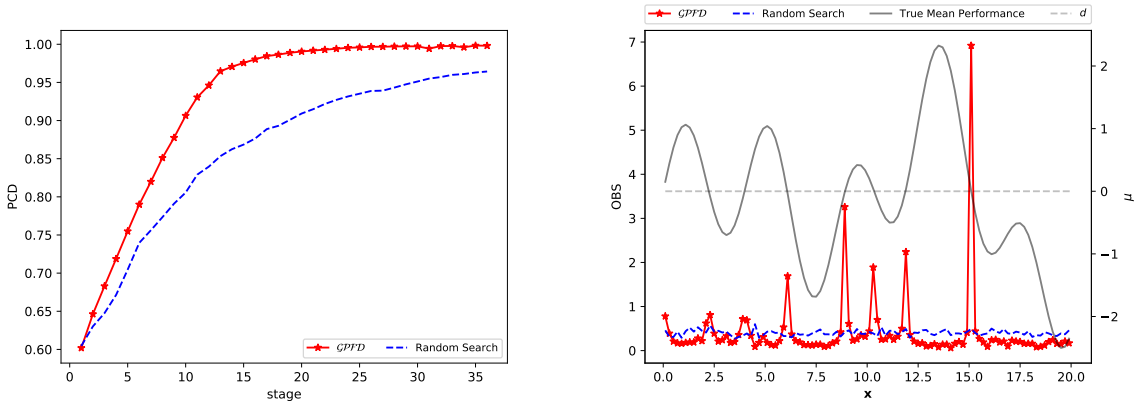


(c) $\sigma_s(\mathbf{x}) = 0.5|\mu(\mathbf{x}) - d|$

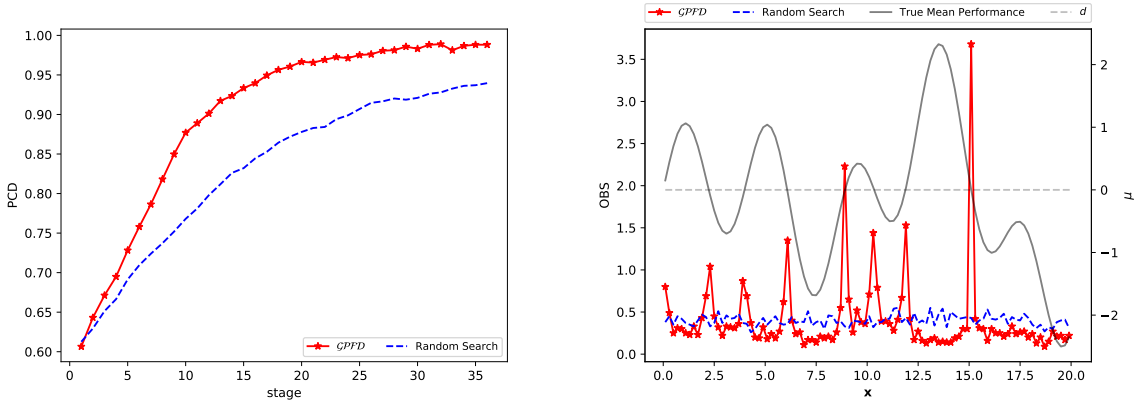
Figure 4.6: One-dimensional experimental results on unequal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = 10$.



(a) $\sigma_s(\mathbf{x}) = 0.1|\mu(\mathbf{x}) - d|$

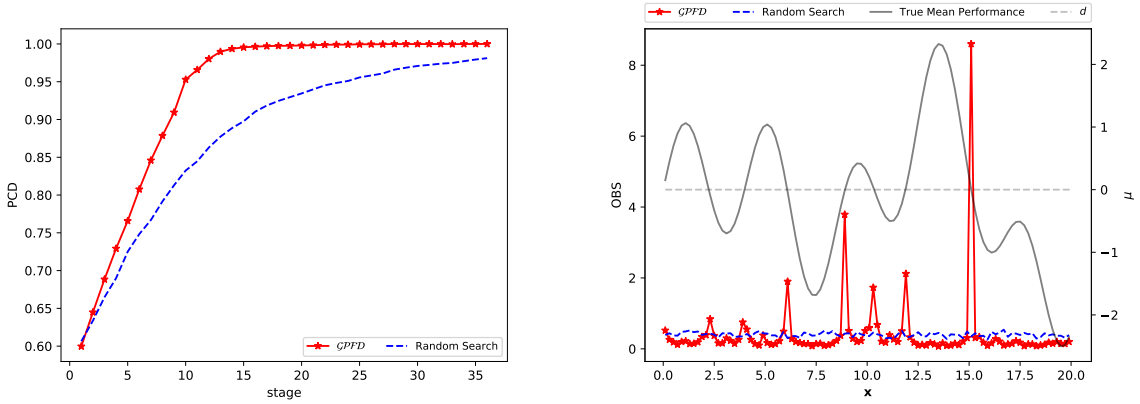


(b) $\sigma_s(\mathbf{x}) = 0.3|\mu(\mathbf{x}) - d|$

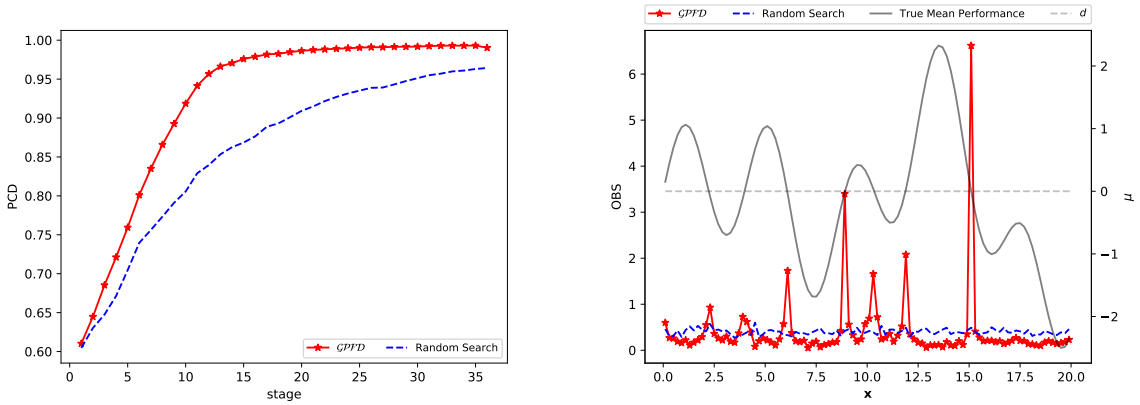


(c) $\sigma_s(\mathbf{x}) = 0.5|\mu(\mathbf{x}) - d|$

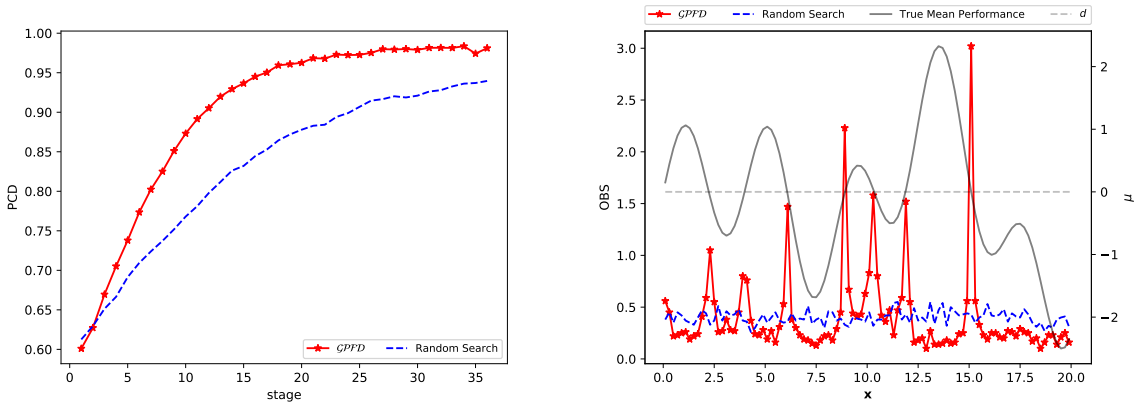
Figure 4.7: One-dimensional experimental results on unequal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = 6$.



(a) $\sigma_s(\mathbf{x}) = 0.1|\mu(\mathbf{x}) - d|$



(b) $\sigma_s(\mathbf{x}) = 0.3|\mu(\mathbf{x}) - d|$



(c) $\sigma_s(\mathbf{x}) = 0.5|\mu(\mathbf{x}) - d|$

Figure 4.8: One-dimensional experimental results on unequal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = 2$.

4.5.2 Two-Dimensional Example

In this example, we test our proposed procedure by using the scaled Goldstein Price function as the true mean performance function. We consider

$$\mathbf{X} = \{(x_1, x_2) : x_1, x_2 \in \{0, 0.02, 0.04, \ldots, 0.98\}\},$$

which consists of 2500 alternatives. Then the true mean performance function is

$$\begin{aligned}\mu(\mathbf{x}) =& \frac{1}{2.427}\Big\{ \log\big(\big[1 + (\bar{x}_1 + \bar{x}_2 + 1)^2 \cdot (19 - 14\bar{x}_1 + 3\bar{x}_1^2 - 14\bar{x}_2 + 6\bar{x}_1\bar{x}_2 + 3\bar{x}_2^2)\big] \\ & \big[30 + (2\bar{x}_1 - 3\bar{x}_2)^2 \cdot (18 - 31\bar{x}_1 + 12\bar{x}_1^2 + 48\bar{x}_2 - 36\bar{x}_1\bar{x}_2 + 27\bar{x}_2^2)\big] \big) - 8.693\Big\},\end{aligned}$$

where $\mathbf{x} = (x_1, x_2)$ and $\bar{x}_1 = 4x_1 - 2, \bar{x}_2 = 4x_2 - 2$. The scaled Goldstein Price function has a mean of 0 and variance 1 on $[0, 1]^2$. For more introduction on the scaled Goldstein Price function, one can refer to [78].

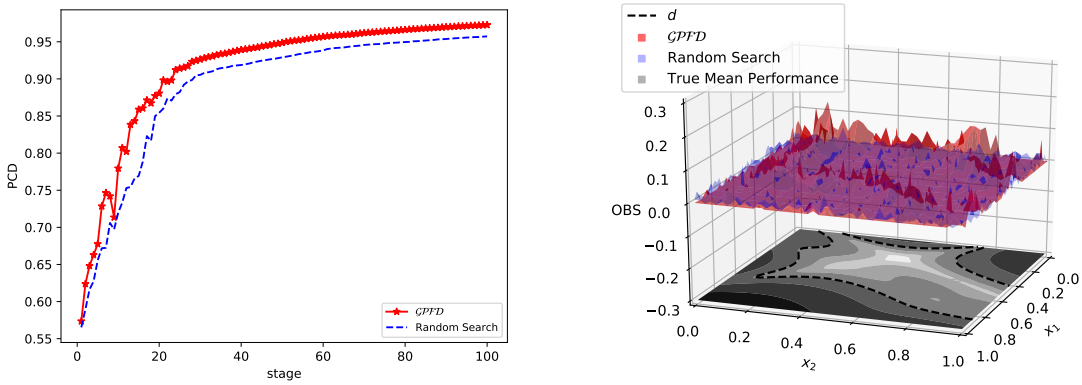
At each stage n, the observation taken is a sample mean of five individual observations from a chosen alternative. The maximum stage is $T = 100$. We want identify all alternatives whose mean performances are less than or equal to $d = 0$. To estimate the prior parameters, we randomly sample four alternatives and take a sample mean of five observations from each of the four sampled alternatives. In this example, we use the accelerated $\mathcal{GPFD}$ procedure but not the full version due to the scale of the available alternatives. We set $w = \sqrt{0.02}$ so that the value of information is computed based on the neighboring 100 alternatives. Similar to the one-dimensional example, we conduct the experiments on both equal and unequal sampling variances cases.

Figures 4.9 and 4.10 show the performances of the accelerated $\mathcal{GPFD}$ procedure under equal and unequal sampling variances cases, respectively. The $\mathcal{GPFD}$ procedure with $w = \sqrt{0.02}$ achieves a final PCD higher than 95% in all cases with just 100 stages (visiting 104 alternatives and obtaining 520 individual observations). Also, it shows higher PCD's

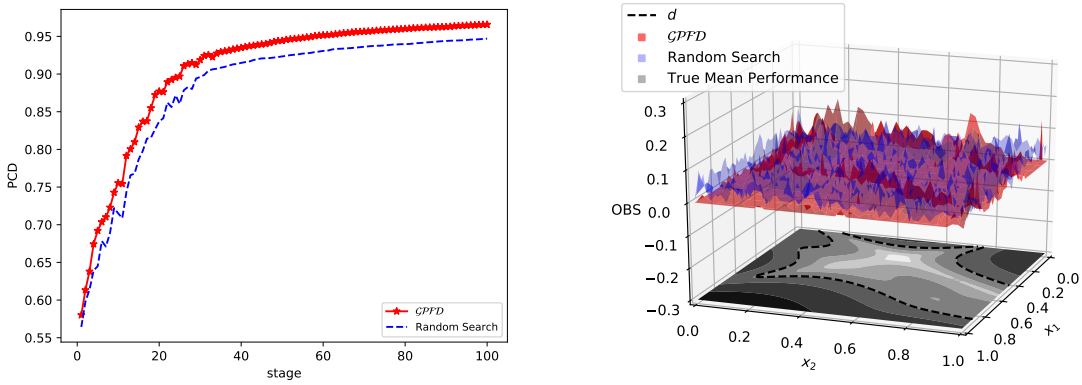
than the random search with GP in all cases, although the difference is smaller than that in the one-dimensional example. The graphs on the right side of Figures 4.9 and 4.10 show that more observations are assigned to barely feasible/infeasible alternatives in the $\mathcal{GPFD}$ procedure.

4.6 Conclusion

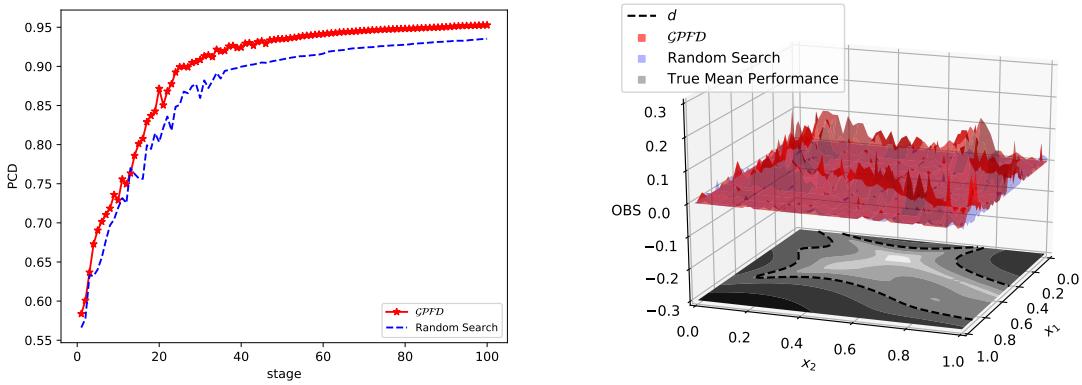
In this chapter, a GP-based sequential procedure for feasibility determination is proposed. The value of information defined as expected increment in reward is used as an acquisition function to decide which alternative to sample next. Two empirical examples are conducted to show that our proposed procedure with the new acquisition function can improve accuracy in finding feasible input regions. For future research, possible directions include extension to a problem with a higher dimensional input space, modeling with a heteroscedastic GP, and using a multi-task GP for multiple performance measures.



(a) $\sigma_s(\mathbf{x}) = 0.1$

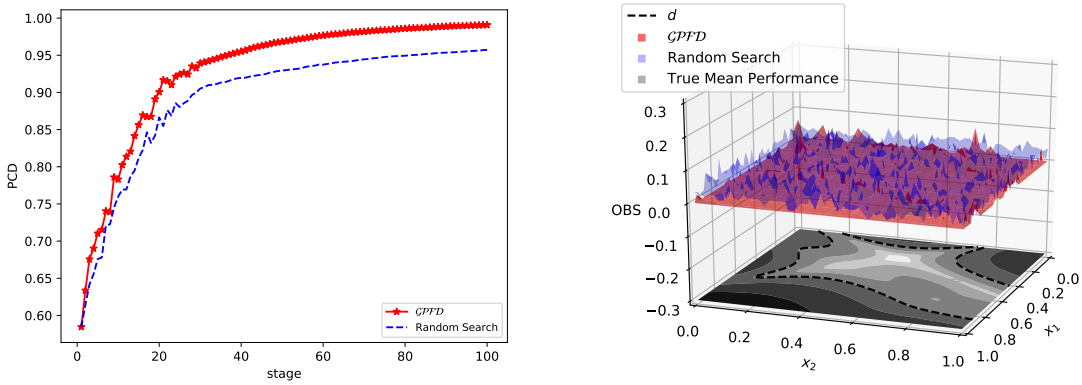


(b) $\sigma_s(\mathbf{x}) = 0.3$

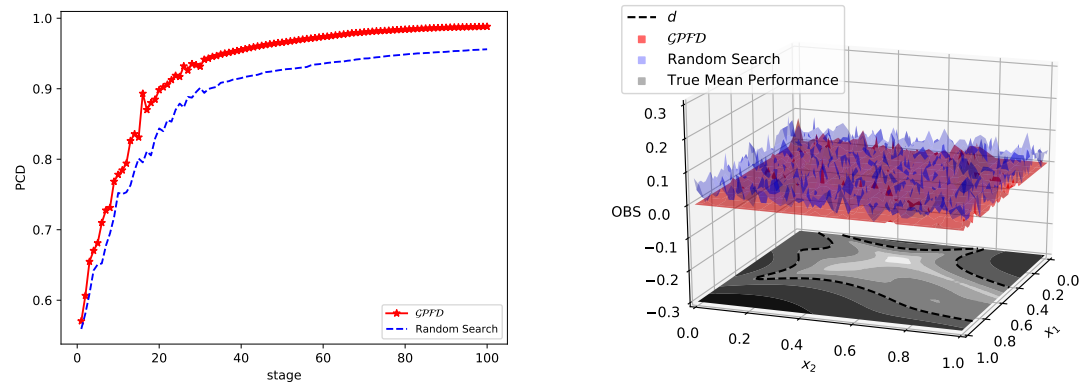


(c) $\sigma_s(\mathbf{x}) = 0.5$

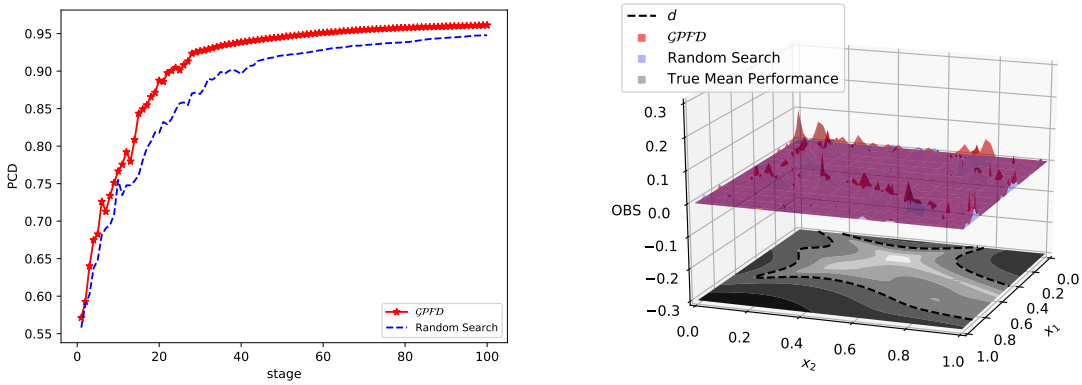
Figure 4.9: Two-dimensional experimental results on equal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = \sqrt{0.02}$.



(a) $\sigma_s(\mathbf{x}) = 0.1|\mu(\mathbf{x}) - d|$



(b) $\sigma_s(\mathbf{x}) = 0.3|\mu(\mathbf{x}) - d|$



(c) $\sigma_s(\mathbf{x}) = 0.5|\mu(\mathbf{x}) - d|$

Figure 4.10: Two-dimensional experimental results on unequal sampling variances when the accelerated version of $\mathcal{GPFD}$ is used and $w = \sqrt{0.02}$.

CHAPTER 5
QUICK AND SITU-AWARE SPATIOTEMPORAL SCHEDULING FOR MANUFACTURING

In this chapter, we provide a general spatiotemporal scheduling formulation and a two-phase approach to the problem. Then we consider the block assembly process from Hyundai Heavy Industries (HHI), the world-largest shipbuilding company in South Korea, and solve their block assembly scheduling problem to demonstrate the efficiency of our proposed method. Note that unlike a typical sequence planning problem where the order of tasks can be switched rather freely, switching the order of tasks in a spatiotemporal scheduling needs to be done carefully considering available bay areas. This spatial scheduling aspect differentiates a spatiotemporal scheduling problem from a classical sequence planning problem. In addition, once a block is placed in a certain location in a bay, the block cannot be moved to a different location until its processing is completed. Thus, a current decision on a block's placement affects future decisions on placements of the other blocks. This temporal scheduling aspect makes the spatiotemporal scheduling problem also different from a classical knapsack problem.

Spatiotemporal scheduling problems are commonly seen in shipbuilding industry, and many related works focus on the block assembly process in shipbuilding. Thus, in the next subsections, we provide more explanation about the shipbuilding process and a literature review of works on the block assembly process in shipbuilding.

A general shipbuilding process is described in Figure 5.1. As a typical production method in modern shipbuilding, a ship is constructed by welding together a number of blocks, which are considered as basic components of a ship. There are mainly three types of blocks: grand assembly, unit assembly and sub-unit assembly blocks. A ship is formed of many grand assembly blocks. Each grand assembly block consists of a number of unit

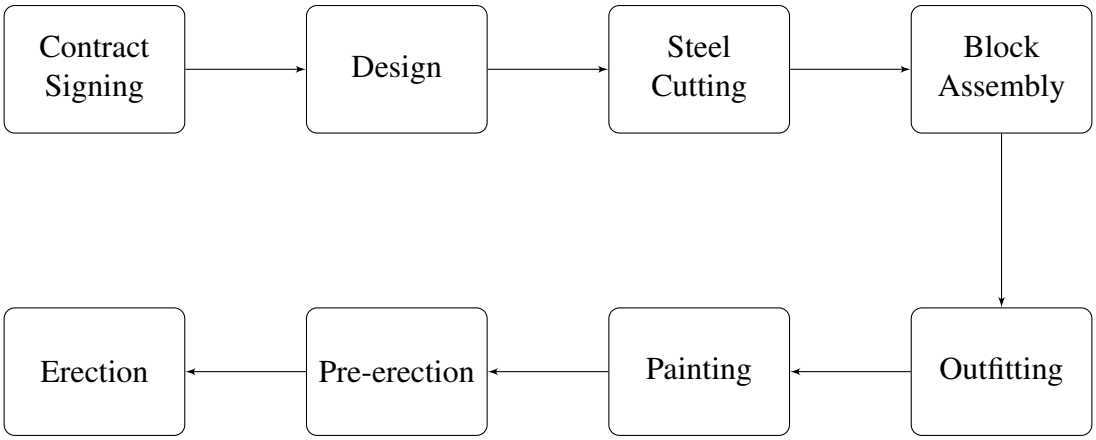


Figure 5.1: Shipbuilding process.

assembly blocks, and each unit assembly block consists of many sub-unit assembly blocks which are made by small parts cut from steel plates. Precedence relations (i.e., a block must be processed after some other blocks) do not exist among grand assembly blocks because they are on top of the construction hierarchy and therefore, can be individually assembled. In this chapter, we consider grand assembly blocks only and we refer to them as blocks for simplicity. Some typical shapes of blocks for shipbuilding are given in Figure 5.2. Blocks are usually assembled in bays. Each bay has its own dimensions and production resources such as cranes and labors. An example of bay layout for a shipbuilding company is shown in Figure 5.3. Each block is given a time window for assembly, and it must be be placed inside an assigned bay to be assembled within this time window. This process is called the block assembly process.

As one of the most difficult and time-consuming processes, the block assembly process is critical in delivering a final product (i.e., a ship) on time. The schedule for the block assembly process should be very carefully planned in order to meet the delivery deadline and optimize production efficiency. Any block that fails to be processed during the given time window due to poor scheduling or capacity limits, which is referred as a failure block, will have to be either delayed or outsourced. There are works that consider delaying the assembly process of failure blocks and some other that consider outsourcing, which are

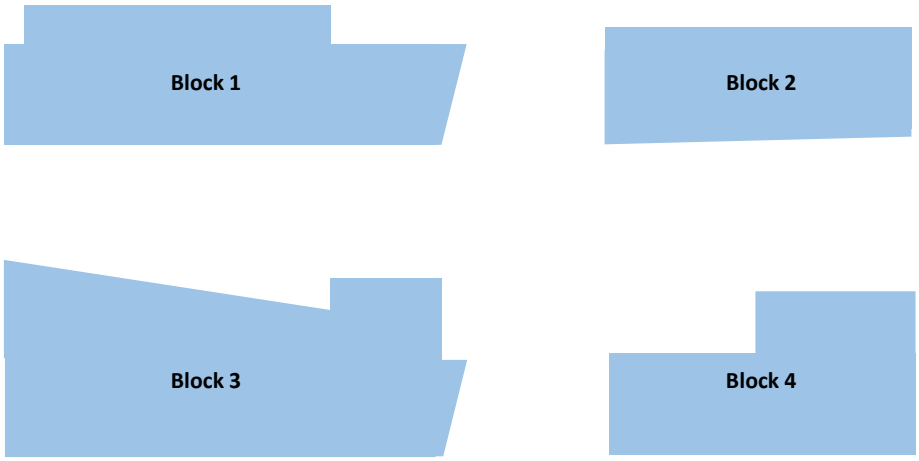


Figure 5.2: An illustrative example of ship blocks.

shown in Section 1.3. In this thesis, we consider that failure blocks will be outsourced to other companies as outside orders.

The problem we consider in this chapter is to find a feasible schedule that tells when and where to assemble each block within fixed time and space constraints. Since any outsourced block will induce an additional cost for the company, one of our main objectives is to minimize the number of failure blocks. In addition, to promote fairness and consistency during the production, we want to balance the workloads across different bays and days.

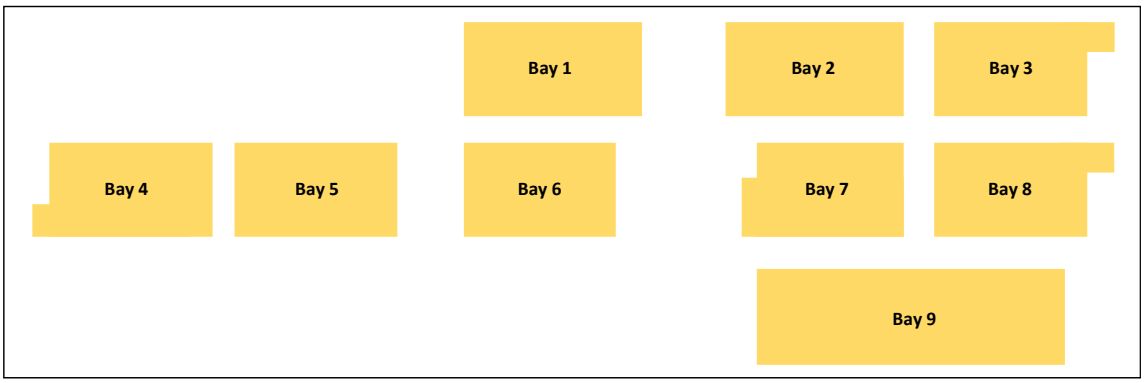


Figure 5.3: An Illustrative example of a bay layout.

Since our goal is not to find a global optimal schedule but to find a reasonably good

one fast, we propose a heuristic two-phase approach to solve the spatiotemporal scheduling problem. Then we test our scheduling procedure using datasets provided by HHI on the block assembly process and compare the resulting schedules with their manual schedules.

This chapter is organized as follows: In Section 5.1, we provide our notation and problem formulation. Then in Section 5.2, we propose our two-phase approach to the problem. Experimental results of the proposed scheduling approach on the datasets from HHI are presented in Section 5.3, followed by conclusions in Section 5.4.

5.1 Notation and Problem

In this section, we provide our notation and problem formulation.

5.1.1 Notation

For a spatiotemporal scheduling problem, let $\boldsymbol{K} = \{1, 2, \ldots, K\}$ be the set of blocks in the schedule, $\boldsymbol{B} = \{0, 1, 2, \ldots, B\}$ be the set of bays, and $\boldsymbol{T} = \{0, 1, 2, \ldots, T\}$ be the set of dates that covers the time horizon of the schedule. Since blocks are coming in and out of bays continuously, there may be blocks that are already under the assembly process at the time when we try to find a new or updated schedule. We refer to these blocks as previously assigned blocks and incoming blocks as unassigned blocks. The set $\boldsymbol{K}$ is essentially the union of previously assigned blocks and unassigned blocks. The time and placement decisions on the previously assigned blocks are fixed to constants, which leave impact on the scheduling of the unassigned blocks due to their occupied bay areas while under assembly.

When a list of blocks to be assembled is received, some initial information is provided such as the initial start date IST(k), initial finish date IFT(k), and assembly duration $D(k)$ for block $k \in \boldsymbol{K}$. The initial start dates are usually determined by the earliest possible start dates for assembly, while the initial finish dates are determined by the deadlines for assembly. A time window to assemble block k is defined by IST(k) and IFT(k). In reality,

the assembly processing times $D(k)$'s are subject to some randomness. However, due to the complexity of the problem, $D(k)$'s are treated as constants and the scheduling problem is modeled as a deterministic optimization problem. Most of the related literature assumes constant assembly processing times as well.

Each block $k \in \boldsymbol{K}$ has width $W(k)$, length $L(k)$, area $A(k)$ and weight $V(k)$. The set $\boldsymbol{R}(k)$ is defined as a set of bays that k cannot be assigned to due to productivity limitation. These given parameters define resource requirements of each block $k \in \boldsymbol{K}$ for its assembly. Although the actual shapes of blocks can be irregular as in Figure 5.2, for simplicity we assume that they are the smallest rectangles that encompass their actual shapes.

Resource capacities for bay $b \in \boldsymbol{B}$ are given by total area TA(b), crane capacity $C(b)$ and a set of available coordinates for placements $\boldsymbol{AC}(b)$.

For each $k \in \boldsymbol{K}$, the decision variable $S(k) = (\text{TD}(k), \text{PD}(k))$ includes both time and placement information, where TD(k) decides when block k is processed and PD(k) determines where it is processed. The time decision variable TD(k) is a single scalar that represents the start date of block k and then the actual finish date becomes TD(k) + $D(k)$, which should be less than or equal to IFT(k). On the other hand, PD(k) is a four-dimensional vector (b, v, h, r) where b is the assigned bay of block k; $(v, h) \in \boldsymbol{AC}(b)$ is the vertical and horizontal coordinates of k's top-left vertex in the assigned bay; and $r \in \{0, 1\}$ indicates the rotation degree with $r = 1$ if rotation degree is 90° or 270° and $r = 0$, otherwise. For convenience, we say that a block is placed with rotation if its rotation degree is 90° or 270°, and without rotation otherwise. For convention, we set TD$(k) = 0$ and PD$(k) = (0, 0, 0, 0)$, if block k is outsourced. Therefore, the set of outsourced blocks can be defined as $\boldsymbol{F} = \{k : \text{TD}(k) = 0 \text{ and } \text{PD}(k) = (0, 0, 0, 0)\}$. Furthermore, we assume that TD(k) and PD(k) are known and fixed, if k is a previously assigned block, because its time and placement information are determined by its current processing progress.

To facilitate modeling of constraints, we define BA(k) as the assigned bay of block k (i.e., the first element in PD(k)), $\boldsymbol{OC}(k)$ as the set of coordinates occupied by block k and

$\boldsymbol{SB}(b,t)$ as the set of blocks in bay b on day t. At the same time, we define $U(b,t)$ as area utilization, which is the proportion of the utilized area over total area of bay b on day t, and $\tilde{U}$ the maximum of $U(b,t)$'s. Finally, the set $\boldsymbol{S} = \{S(1), S(2), \cdots, S(K)\}$ is the schedule for the whole process.

Full descriptions about the notation defined in this section can be found in Table 5.1.

Table 5.1: Summary of notation.

	notation	meaning
parameters	$\boldsymbol{K}$	set of blocks in the schedule, $\boldsymbol{K} = \{1, 2, \ldots, K\}$.
	$\boldsymbol{B}$	set of available bays, $\boldsymbol{B} = \{0, 1, \ldots, B\}$.
	$\boldsymbol{T}$	set of dates in the time horizon, $\boldsymbol{T} = \{0, 1, \ldots, T\}$.
indices	k	index of blocks, $k \in \boldsymbol{K}$.
	b	index of bays, $b \in \boldsymbol{B}$.
	t	index of dates, $t \in \boldsymbol{T}$.
resource requirements	IST(k)	initial start date to assemble k.
	IFT(k)	initial finish date to assemble k.
	$D(k)$	processing duration of k.
	$W(k)$	width of k in meters.
	$L(k)$	length of k in meters.
	$A(k)$	area of k in squared meters.
	$V(k)$	weight of k in tons.
	$\boldsymbol{R}(k)$	set of bays that k cannot be assigned to.
resource capacities	TA(b)	total area of b in squared meters.
	$\boldsymbol{AC}(b)$	set of available coordinates for placements in bay b in meters.
	$C(b)$	crane capacity in bay b in tons.
decision variables	TD(k)	time decision on block k, which represents the start date when k is assembled.
	PD(k)	placement decision on block k, PD$(k) = (b, v, h, r)$ where b is the assigned bay of k, (v, h) are the vertical and horizontal coordinates of its left-top vertex, and r is the rotation indicator.
	$S(k)$	schedule decision on k, $S(k) = (\text{TD}(k), \text{PD}(k))$.
	$\boldsymbol{S}$	set of all schedule decisions $\boldsymbol{S} = \{S(1), S(2), \cdots, S(K)\}$.
auxiliary variables	$\boldsymbol{F}$	set of outsourced blocks, $\{k : \text{TD}(k) = 0 \text{ and PD}(k) = (0,0,0,0)\}$.
	BA(k)	assigned bay of k (the first element in PD$(k) = (b, v, h, r)$).
	$\boldsymbol{OC}(k)$	set of coordinates occupied by k (determined by PD(k)).
	$\boldsymbol{SB}(b,t)$	set of blocks in b on t (determined by TD(k) and PD(k)).
	$U(b,t)$	utilization of bay b on day t, which is the proportion of utilized area over total area.
	$\tilde{U}$	maximum of $U(b,t)$, $b \in \boldsymbol{B}$, $t \in \boldsymbol{T}$.

5.1.2 Problem Formulation

A function $\mathcal{M} : \boldsymbol{S} \to \mathbb{R}$ is a function of $\boldsymbol{S}$ and represents certain performance measure of the schedule. Examples of popular functions $\mathcal{M}$'s are the time span of the assembly process, the number of outsourced blocks, fluctuation of workloads distribution and so on. The spatiotemporal scheduling problem can be formulated as follows:

$$
\begin{aligned}
\min_{\boldsymbol{S}} \quad & \mathcal{M}(\boldsymbol{S}), \\
\text{s.t.} \quad & \text{time window constraints;} \\
& \text{placement constraints;} \\
& \text{production factor constraints;} \\
& \text{non-overlapping constraints;} \\
& \text{specialty constraints;} \\
& \text{additional constraints.}
\end{aligned}
$$

As mentioned earlier, the performance measures under consideration are the number of failure blocks and discrepancy of workloads across bays and days. The discrepancy of workloads is reflected by the maximum of utilizations, because a smaller maximum of utilization leads to more balanced workloads under the same total amount of workloads. Consequently, we set the function $\mathcal{M}(\boldsymbol{S})$ as a combined measurement of $|\boldsymbol{F}|$ and $\tilde{U}$ and the objective is to minimize $\mathcal{M}(\boldsymbol{S})$.

Many types of constraints should be considered in this scheduling problem. The relevant constraints are stated as follows:

- Time window constraints: for each $k \in \boldsymbol{K} \setminus \boldsymbol{F}$,

$$\text{TD}(k) \geq \text{IST}(k), \tag{5.1}$$

$$\text{TD}(k) + D(k) \ \leq \text{IFT}(k). \tag{5.2}$$

- Placement constraints: for each $k \in \boldsymbol{K} \setminus \boldsymbol{F}$,

$$\boldsymbol{OC}(k) \subseteq \boldsymbol{AC}\left(\text{BA}(k)\right). \tag{5.3}$$

- Production factor constraints: for each $b \in \boldsymbol{B}$ and each $t \in \boldsymbol{T}$,

$$\sum_{k \in \boldsymbol{SB}(b,t)} A(k) \leq \text{TA}(b,t), \tag{5.4}$$

$$\max_{k \in \boldsymbol{SB}(b,t)} V(k) \leq C(b). \tag{5.5}$$

- Non-overlapping constraints: for each pair (k_1, k_2) such that $k_1, k_2 \in \boldsymbol{SB}(b,t)$ and $k_1 \neq k_2$, $b \in \boldsymbol{B}$ and $t \in \boldsymbol{T}$,

$$\boldsymbol{OC}(k_1) \cap \boldsymbol{OC}(k_2) = \emptyset. \tag{5.6}$$

- Specialty constraints: for each $k \in \boldsymbol{K} \setminus \boldsymbol{F}$,

$$\text{BA}(k) \in \boldsymbol{B} \setminus \boldsymbol{R}(k). \tag{5.7}$$

- Additional constraints: for each $b \in \boldsymbol{B}$ and each $t \in \boldsymbol{T}$,

$$U(b,t) \leq \tilde{U}, \text{ for each } b \in \boldsymbol{B} \text{ and } t \in \boldsymbol{T}, \tag{5.8}$$

$$U(b,t) = \sum_{k \in \boldsymbol{SB}(b,t)} \frac{A(k)}{\text{TA}(b)}, \text{ for each } b \in \boldsymbol{B} \text{ and } t \in \boldsymbol{T}. \tag{5.9}$$

Finding an optimal solution to the above spatiotemporal scheduling problem is NP-hard [79]. Instead, we propose a two-phase approach that finds a good solution fast in the next section.

5.2 Two-Phase Approach

In this section, we present our approach in finding a good solution to the spatiotemporal scheduling problem fast. Our approach divides the whole scheduling problem into two phases. In Phase I, it finds an intermediate schedule that decides the start date TD(k) and assigned bay BA(k) (or the first element of PD(k)) for each block. Phase I considers areas of blocks and bays only and a block can be placed in a bay as long as the bay has enough available area. Then in Phase II, we further specify the coordinates and rotations to place the blocks based on the results from Phase I, which determines the remaining elements in PD(k). Since Phase II considers the dimensions of blocks and bays, it is possible that not all blocks assigned to a bay in Phase I can be eventually placed in the bay in Phase II.

5.2.1 Phase I

In Phase I, we want to decide when each block is assembled, which bay it is assigned to, and which blocks are outsourced in this phase. In other words, the decisions to be made are TD(k), BA(k) and $\boldsymbol{F}_I$, where $\boldsymbol{F}_I$ is the set of blocks to be outsourced in Phase I. For the objective in the model, we want to minimize $|\boldsymbol{F}_I| + \tilde{U}$, which assigns the same weights of 1 to both performance measures $|\boldsymbol{F}_I|$ and $\tilde{U}$. As $\tilde{U}$ is always less than 1, the performance measure $\boldsymbol{F}_I$ dominates the quality of a solution. This choice of weights is reasonable, because the number of failure blocks is the main cause of additional cost and it should be prioritized. When there are solutions with the same value of $\boldsymbol{F}_I$, $\tilde{U}$ can distinguish which one we prefer further, because a smaller maximum of utilizations leads to more balanced workloads. The problem in Phase I is as follows:

$$\begin{aligned} \min \quad & |\boldsymbol{F}_I| + \tilde{U}, \\ \text{s.t.} \quad & (5.1), (5.2), (5.4), (5.5), (5.7), (5.8), (5.9). \end{aligned}$$

Table 5.2: Additional notation for Phase I.

	notation	meaning
parameters	$I(k,b)$	1 if block k is allowed to be assigned to bay b; 0 otherwise.
	$P(k,s,t)$	1 if block k has start date s and it is still under assembly on day t; 0 otherwise.
	$\tilde{d}(k,s,b)$	1 if a previously assigned block k has start date s and assigned bay b; 0 otherwise.
	$\boldsymbol{PA}$	set of blocks that are previously assigned.
	$\boldsymbol{ST}(k)$	set of all possible start dates to assemble block k.
decision variables	$d(k,s,b)$	1 if block k has start date s and assigned bay b; 0 otherwise.
	$\boldsymbol{F}_I$	set of outsourced blocks in Phase I.

Note that an optimization solver cannot solve the above problem formulation directly but needs to be converted into a mixed integer programming model. We need additional notation for the integer programming model. A set of all possible start dates to assemble block k is $\boldsymbol{ST}(k)$. Possible assembly schedules for block k are specified by $P(k,s,t)$'s. If block k's assembly starts on day $s \in \boldsymbol{ST}(k)$ and it is still under processing on day t, then $P(k,s,t) = 1$. Otherwise, it takes the value of zero. For example, suppose that block k's earliest assembly day is day 1 and latest assembly finish day is day 4 with three days of processing time (i.e., IST$(k) = 1$, IFT$(k) = 4$ and $D(k) = 3$). Then $\boldsymbol{ST}(k) = \{1,2\}$, $P(k,1,1) = P(k,1,2) = P(k,1,3) = 1$, $P(k,2,2) = P(k,2,3) = P(k,2,4) = 1$ and the remaining $P(k,s,t)$'s are zeros. In Section 5.1, $\boldsymbol{B} \setminus \boldsymbol{R}(k)$ denotes a set of bays where block k can be assigned. For the mixed integer programming, we use $I(k,b)$ which takes 1 if block k can be assigned to bay b, and 0 otherwise. Additional notation defined for Phase I is given in Table 5.2.

We denote the mixed-integer programming model as M_I and the full description of M_I is given in Appendix C.

5.2.2 Phase II

While Phase I finds TD(k) and BA(k) (or the first element of PD(k)), the goal of Phase II is to assign coordinates and rotation degrees to place the blocks using the intermediate

schedule from Phase I. As Phase II considers dimensions of blocks and bays while Phase I considers only their areas, it is possible that additional blocks need to be outsourced in this phase. Let $\boldsymbol{F}_{II}$ represent the set of blocks to be outsourced in Phase II. Then we want to find the remaining elements of PD(k) and decide $\boldsymbol{F}_{II}$. Assuming that we have an intermediate schedule from Phase I, the Phase II problem is as follows:

$$\begin{aligned} \min \quad & |\boldsymbol{F}_{II}|, \\ \text{s.t.} \quad & (5.3), (5.6). \end{aligned}$$

Since the placements of blocks in each bay are independent of those in the other bays, we can break the complex problem of placement assignments into subproblems, each of which involves only one bay. Then, using parallel computing, all the subproblems can be solved at the same time on separate processors, which largely reduces the total run time of the scheduling process. From this point, we consider the placement assignments of bay b.

Although our goal is to find coordinates and rotation degrees for placing assigned blocks in each bay over the whole time horizon $\boldsymbol{T}$, it is not necessary to work on placements on all $t \in \boldsymbol{T}$. For some $t \in \boldsymbol{T}$, the set of blocks may be a subset of some other day $t' \in \boldsymbol{T}$ (i.e.,$\boldsymbol{SB}(b,t) \subseteq \boldsymbol{SB}(b,t')$). Therefore, we only need to look for placement assignments on a subset of $\boldsymbol{T}$ defined as $\boldsymbol{TT}(b) = \{t \in \boldsymbol{T} : \boldsymbol{SB}(b,t) \not\subseteq \boldsymbol{SB}(b,t'), \forall t' \in \boldsymbol{T} \text{ and } t' \neq t\}$. Once the placements on each $t \in \boldsymbol{TT}(b)$ are decided, the placements on the other days are decided as well.

To leave more usable area for future blocks, we try to place blocks to the corner as much as possible. Therefore, the objective is to maximize the distances from the centers of the placed blocks to the center of the bay. To express this objective and represent constraints (5.3) and (5.6) in an integer programming model, we need additional notation which is given in Table 5.3. The descriptions of a mixed-integer programming model for Phase II, denoted by M_{II}, is given in Appendix C.

To solve the placement problem on each bay, we use a heuristic approach. Let $t^{(i)}$ be

Table 5.3: Phase II model notation.

	notation	meaning
parameters	$\tilde{a}(k, v, h)$	1 if a previously assigned block k is placed at (v, h) without rotation; 0 otherwise.
	$\tilde{a}_r(k, v, h)$	1 if a previously assigned block k is placed at (v, h) with rotation; 0 otherwise.
	$w(k, v, h)$	distance from the center of k to the center of its assigned bay if k's left-top vertex is placed (v, h) without rotation.
	$w_r(k, v, h)$	distance from the center of k to the center of its assigned bay if k's left-top vertex is placed (v, h) with rotation.
	$\boldsymbol{TT}(b)$	set of days when coordinate assignments need to be done.
	$\boldsymbol{SC}(k)$	set of coordinates that k's left-top vertex can possibly be without rotation.
	$\boldsymbol{SC}_r(k)$	set of coordinates that k's left-top vertex can possibly be with rotation.
	$\boldsymbol{NA}(b, t)$	set of blocks that are newly assigned to bay b on day t by the intermediate schedule from Phase I.
	$\boldsymbol{PA}(b, t)$	set of blocks that are previous assigned to bay b on day t.
decision variables	$\boldsymbol{F}_{II}$	set of outsourced blocks in Phase II.
	$a(k, v, h)$	1 if k's left-top vertex is at (v, h) without rotation; 0 otherwise.
	$a_r(k, v, h)$	1 if k's left-top vertex is at (v, h) with rotation; 0 otherwise.
auxiliary variables	$hc(k_1, k_2)$	1 if k_1 is placed without rotation to the left of k_2 and they are non-overlapping horizontally; 0 otherwise.
	$hc_r(k_1, k_2)$	1 if k_1 is placed with rotation to the left of k_2 and they are non-overlapping horizontally; 0 otherwise.
	$vc(k_1, k_2)$	1 if k_1 is placed without rotation to the top of k_2 and they are non-overlapping vertically; 0 otherwise.
	$vc_r(k_1, k_2)$	1 if k_1 is placed with rotation to the top of k_2 and they are non-overlapping vertically; 0 otherwise.

the ith earliest day in $\boldsymbol{TT}(b)$. Starting from $i = 1$, we solve M_{II} in Appendix C and move on to next i until $i = |\boldsymbol{TT}(b)|$. The whole Phase II algorithm is then stated in Algorithm 5.

Algorithm 5

1: **Initialization**: Prepare all parameters in Table 5.3. Specify the target bay b in which placement decisions are made. Let $\boldsymbol{F}_{II}(b) = \emptyset$. Set $i = 1$.
2: **Optimization**: Solve M_{II} for $t = t^{(i)}$. If a solution is found, go to **Update**. Otherwise, go to **Removal**.
3: **Update**: Fix all decision variables $a(k, v, h)$ and $a_r(k, v, h)$ to the current solution for all $k \in \boldsymbol{SB}(b, t)$. If $i = |\boldsymbol{TT}(b)|$, go to **Termination**. Otherwise, set $i = i + 1$ and go to **Optimization**.
4: **Removal**: Remove k that has the latest start date in $\boldsymbol{NA}(b, t)$ from both $\boldsymbol{NA}(b, t)$ and $\boldsymbol{SB}(b, t)$. Add k to $\boldsymbol{F}_{II}(b)$. Go to **Optimization**.
5: **Termination**: Return all decision variables $a(k, v, h)$ and $a_r(k, v, h)$ and $\boldsymbol{F}_{II}(b)$.

After solving all the placement problems for all $b \in \boldsymbol{B}$, we can fully determine $\mathbf{PD}(k)$ for all $k \in \boldsymbol{K}$ and obtain $\boldsymbol{F}_{II} = \boldsymbol{F}_{II}(1) \cup \cdots \cup \boldsymbol{F}_{II}(B)$ and $\boldsymbol{F} = \boldsymbol{F}_I \cup \boldsymbol{F}_{II}$.

5.3 Experiments

In this section, we compare the automated schedules resulting from our two-phase approach with the manual schedules provided by HHI. The performance metrics for comparison are defined as follows:

- ESD: total earliness in start dates of the assigned blocks, $\sum_{k \in \boldsymbol{K}} \max(0, \text{IST}(k) - \text{TD}(k))$.
- LFD: total lateness in finish dates of the assigned blocks, $\sum_{k \in \boldsymbol{K}} \max(0, \text{TD}(k) + D(k) - \text{IFT}(k))$ where $\text{TD}(k) + D(k)$ is the finish date of block k.
- OL10: the number of blocks that are overlapped by the other for more than 10% of their areas for more than one day.
- OL30: the number of blocks that are overlapped by the other for more than 30% of their areas for more than one day.

- OB: the number of blocks that are out of the boundaries of their assigned bays.
- MU: maximum of monthly average utilizations across all bays, which is computed by measuring all the monthly average utilizations of all the bays and then taking the maximum of them.
- $|\boldsymbol{F}|$: for the manual schedules, this quantity is measured by the number of blocks that need to be removed in an inverse chronological order so that there is no more block overlapped for more than 30% of its area for more than one day.

These performance metrics are considered as valuable by the manufacturer, as they indicate how smooth and robust a schedule is.

Six datasets are considered in the experiment, each with 637–810 blocks (i.e., the number of blocks K is between 637 and 810) and a six-month planning horizons. Each dataset represents one scheduling period, and contains blocks that are already under the assembly process and incoming blocks that have not been scheduled. For each dataset (or scheduling period), the assembly schedules of the previously assigned blocks are already determined and thus, their time and placement decision variables are fixed during the current scheduling period. These datasets simulate the actual working situation of the shipbuilding company, in which blocks keeps coming in and out for assembly.

5.3.1 Comparison Results

The comparison results are shown in Table 5.4. For automated schedules, we do not expect any earliness, lateness, overlapped blocks or out-of-boundary blocks, while manual schedules have lateness and a good number of overlapped blocks and a few out-of-boundary blocks. Both types of schedules show similar maximum monthly average utilization, but automated schedules tend to have a smaller number of outsourced blocks. The most outstanding saving is the computation time in finding a schedule. It takes about 2 weeks to come up with a manual schedule for each dataset. However, for an automated schedule, it

takes about 2 hours if Phase II is run sequentially one bay after another, or about 30 minutes if Phase II is implemented with parallel computing.

Table 5.4: Summary of comparison results.

		Manual Schedules							**Automated Schedules**						
Dataset	K	ESD	LFD	OL10	OL30	OB	MU	$\lvert\boldsymbol{F}\rvert$	ESD	LFD	OL10	OL30	OB	MU	$\lvert\boldsymbol{F}\rvert$
data 1	637	0	0	38	9	0	0.625	9	0	0	0	0	0	0.617	6
data 2	641	0	0	30	4	0	0.679	4	0	0	0	0	0	0.644	8
data 3	810	0	1	37	11	0	0.641	10	0	0	0	0	0	0.660	6
data 4	755	0	5	26	3	0	0.632	2	0	0	0	0	0	0.620	4
data 5	728	0	10	40	10	2	0.623	10	0	0	0	0	0	0.627	5
data 6	727	0	9	38	7	3	0.626	6	0	0	0	0	0	0.621	2
Average Computation Time		$\approx$ 2 weeks							$\approx$ 2 hours						

5.3.2 Sample Block Placements

We present two small samples of block placements resulting from the manual schedules and the automated schedules, respectively. Both samples are based on the same dataset, and they are the block placements on the same bay from the same time range 2019/09/20 to 2019/10/04 (10 workdays). The sample of manual schedules is shown in Figure 5.4 and the one of automated schedules is in Figure 5.5. The area inscribed by the red solid lines is the possible area for block assembly. From the figures, we can see that the manual schedules have infeasible placements that place some blocks over the red solid lines. As the red solid lines represent the definite boundaries of the bay, it is impossible to place blocks over the red solid lines. On the other hand, the automated schedules guarantee to have all valid placements.

5.4 Conclusion

In this chapter, we present a heuristic two-phase approach for a spatiotemporal scheduling problem that has efficient running time and can adapt changes in the progress status to obtain a new updated schedule. The two-phase approach is applied to a block assembly scheduling problem that arises in the block assembly process in shipbuilding. From the experimental results based on the datasets provided by HHI, we see that our scheduling

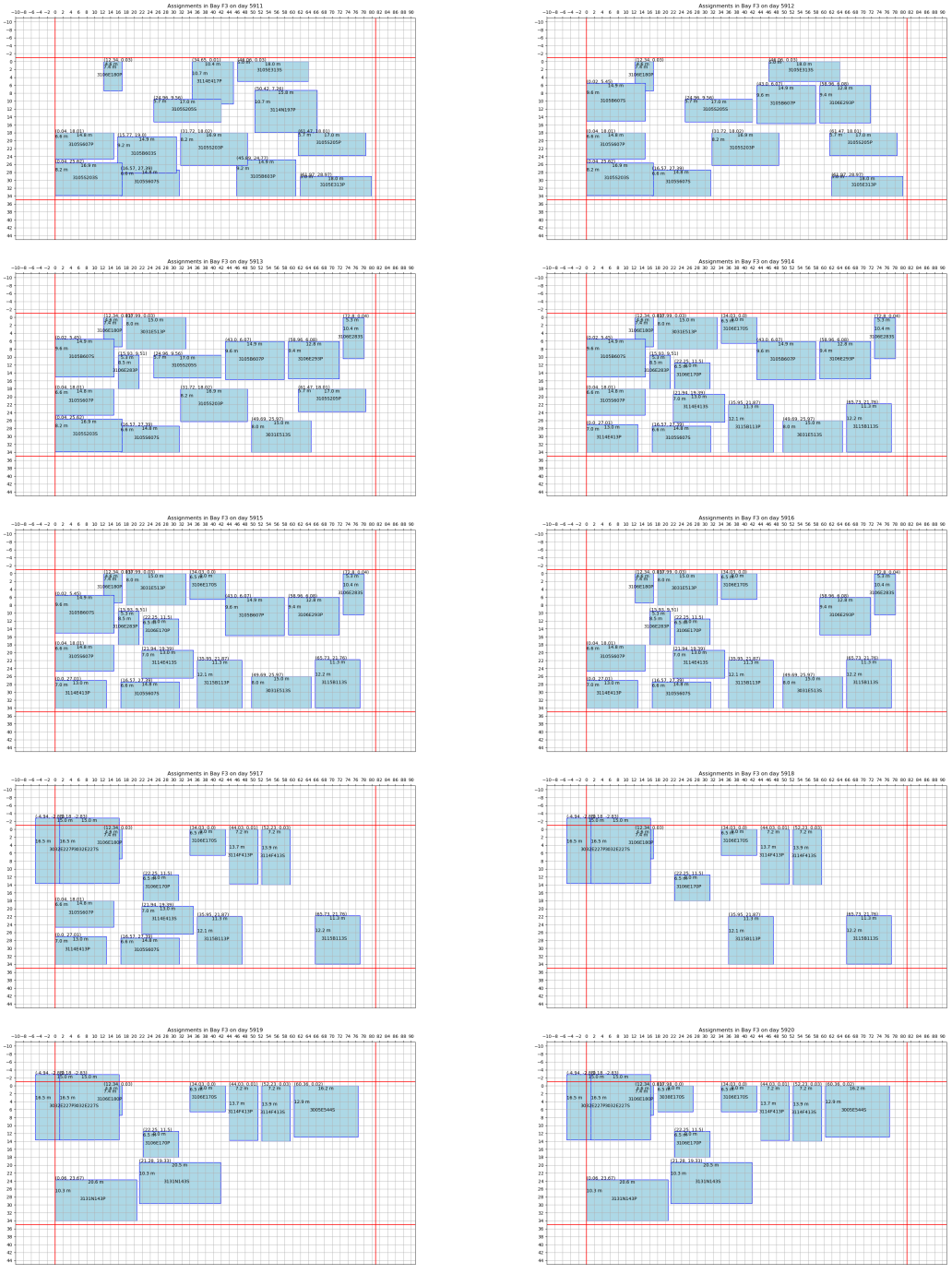


Figure 5.4: A sample of block placements from manual schedules for 10 days. Figures are placed in chronological order from left to right starting from the top.

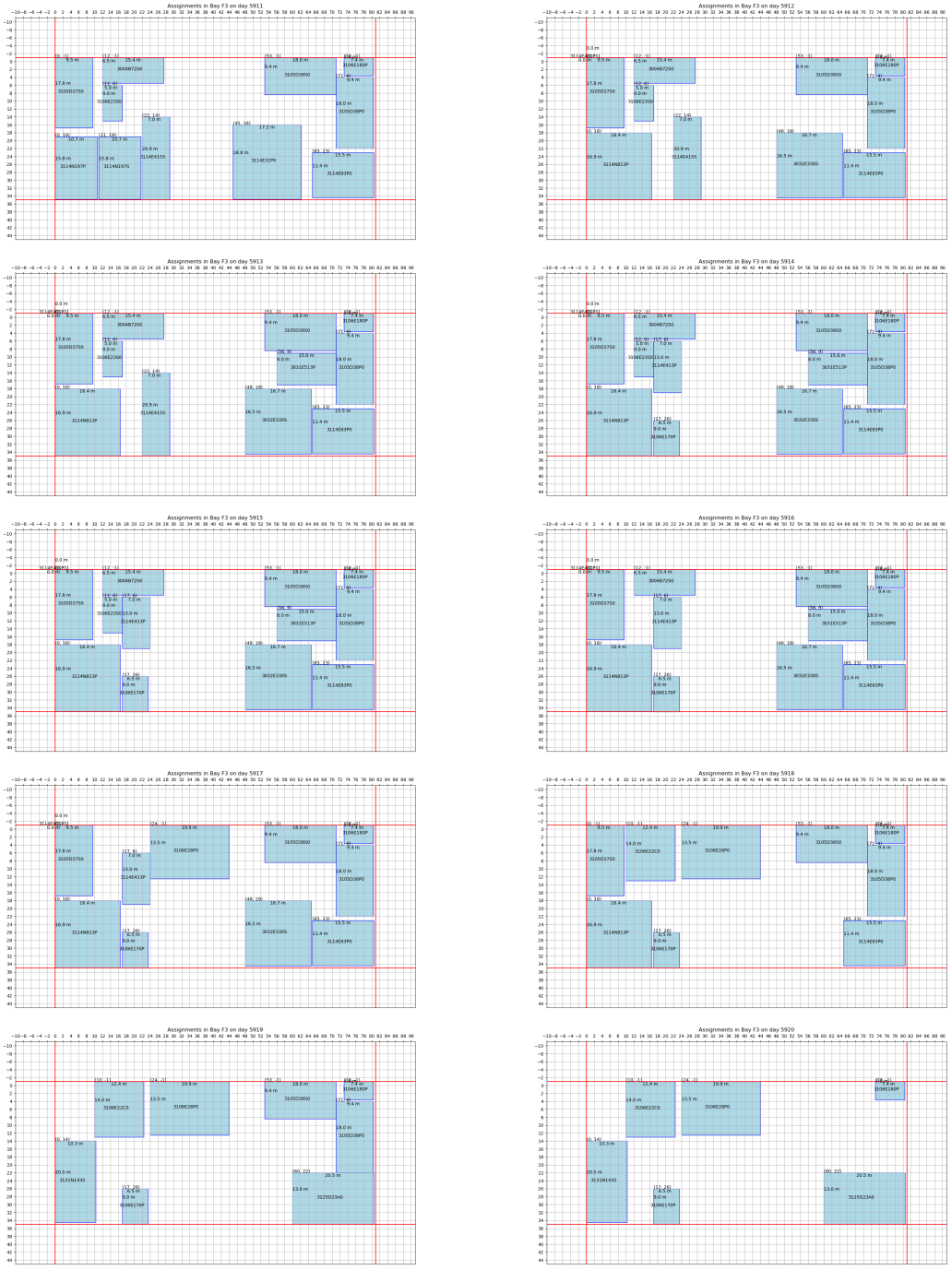


Figure 5.5: A sample of block placements from automated schedules. Figures are placed in chronological order from left to right starting from the top.

procedure generates better schedules with much shorter computation time than HHI's manual schedules. More specifically, our schedules satisfy all necessary constraints for blocks assigned to bays, while keeping the number of outsourced blocks smaller than that from the manual schedule in most testing cases. Our approach also largely reduces the average computation time for generating a schedule. Note that our models can take the status of blocks currently undergoing the assembly process as prior information before making schedules for future unscheduled blocks. Therefore, our scheduling procedure is situ-aware in the sense that whenever there is a change in the assembly process, such as delay in arrival dates of blocks, longer assembly times than expected, change of a time horizon, etc, one can rerun the two-phase approach with updated information and get a new schedule in a relatively short time. However, our current approach assumes deterministic parameters, and therefore, the resulting schedule may not be robust if there exists a high level of uncertainty in, for example, assembly durations.

CHAPTER 6
FUTURE RESEARCH

In Chapter 2, we have presented a new type of loss/reward function that has the shape of a normal distribution and in Chapter 3, we have developed new Bayesian procedures that are more effective than the chosen benchmark procedures for solving the feasibility determination problem. We have presented a GP-based feasibility determination procedure with a new acquisition function in Chapter 4.

Although our proposed procedures have shown promising performances, there are still many limitations. First of all, all our procedures are designed for a single constraint only. Extension to multiple constraints is nontrivial due to the difficulty of getting an explicit expression of the expected Bayesian reward/loss or the acquisition function for a multi-task GP, when different performance measures are dependent on each other. However, many applications may concern multiple constraints on the performance measures that are mutually dependent. For example, an inventory manager wants to use simulation to identify inventory policies that return an order fulfillment rate higher than 90% and operational cost lower than 1 million dollars per month. Second, our procedures assume that the number of available systems for simulation is finite. In some simulation environments, the input parameters that control the simulation output are continuous, and therefore, there are infinitely many different systems. For example, a facility manager wants to use simulation to find airflow amounts that keep the facility's temperature lower than $50°$C. Both limitations provide potential directions for future study.

Finally, a fast heuristic algorithm is presented for spatiotemporal scheduling in manufacturing in Chapter 5. We have applied the algorithm to blocking scheduling data from HHI and were able to reduce the scheduling time from two weeks to two hours. However, the algorithm assumes deterministic parameters and ignore uncertainties in factors such as

block arrival times, processing times and available human labor hours, which is a major limitation. As a result, one possible future project is to consider stochasticity in solving the scheduling problem.

Appendices

APPENDIX A

SUPPORTING MATERIALS FOR CHAPTER II

In this appendix, we show the technical details related to the sampling procedures presented in Chapter 2.

A.1 Proofs that the Normal Reward Function Satisfies Conditions 1, 2 and 3

First, we state the following lemmas:

Lemma 1. *Define* $G(\eta_f, \eta_g, \lambda_f, \lambda_g) = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{\lambda_f \lambda_g}{\lambda_f + \lambda_g}} \exp\left\{-\frac{1}{2}(\eta_f - \eta_g)^2 \frac{\lambda_f \lambda_g}{\lambda_f + \lambda_g}\right\}$.

Then,

$$\mathbb{E}\left[G(d, \eta_{n,i}, b, \lambda_{n,i}) | S_{0,i} = (\eta, \lambda)\right] = G(d, \eta, b, \lambda),$$

where $S_{n,i} = (\eta_{n,i}, \lambda_{n,i})$.

Proof. We first prove the case where $n = 1$:

$$
\begin{aligned}
&\mathbb{E}\left[G(d, \eta_{1,i}, b, \lambda_{1,i}) | S_{0,i} = (\eta, \lambda)\right] \\
=&\int_{-\infty}^{\infty} G(d, \eta + \tilde{\sigma}_i(\lambda) \cdot z, b, \lambda + \gamma_i) \cdot \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2} z^2\right\} dz \\
=&\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b(\lambda+\gamma_i)}{b+\lambda+\gamma_i}} \exp\left\{-\frac{1}{2}(d - (\eta + \tilde{\sigma}_i(\lambda) \cdot z))^2 \frac{b(\lambda+\gamma_i)}{b+\lambda+\gamma_i}\right\} \\
&\cdot \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2} z^2\right\} dz \\
=&\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} \int_{-\infty}^{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}} \exp\left\{-\frac{1}{2}(z - \frac{d-\eta}{\tilde{\sigma}_i(\lambda)})^2 \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}\right\} \\
&\cdot \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2} z^2\right\} dz \\
=&\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} \cdot G\left(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, 0, \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}, 1\right) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)} + 1} \\
&\cdot \exp\left\{-\frac{1}{2}(z - \frac{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)} b\gamma_i}{b\gamma_i + \lambda(b+\lambda+\gamma_i)})^2 \cdot (\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)} + 1)\right\} dz \\
=&\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} G\left(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, 0, \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}, 1\right) \\
=&\frac{1}{\sqrt{2\pi}} \sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} \cdot \sqrt{\frac{b\gamma_i}{b\gamma_i + \lambda(b+\lambda+\gamma_i)}} \\
&\cdot \exp\left\{-\frac{1}{2}(d-\eta)^2 \cdot \frac{\lambda(\lambda+\gamma_i)}{\gamma_i} \cdot \frac{b\gamma_i}{b\gamma_i + \lambda(b+\lambda+\gamma_i)}\right\} \\
=&\frac{1}{\sqrt{2\pi}} \sqrt{\frac{b\lambda}{b+\lambda}} \cdot \exp\left\{-\frac{1}{2}(d-\eta)^2 \cdot \frac{b\lambda}{b+\lambda}\right\} \\
=&G(d, \eta, b, \lambda).
\end{aligned}
$$

By Tower rule,

$$
\begin{aligned}
\mathbb{E}\left[G(d, \eta_{n,i}, b, \lambda_{n,i}) | S_{0,i} = (\eta, \lambda)\right] &= \mathbb{E}\left[G(d, \eta_{n-1,i}, b, \lambda_{n-1,i}) | S_{0,i} = (\eta, \lambda)\right] \\
&= \ldots = G(d, \eta, b, \lambda),
\end{aligned}
$$

for any stage $n \geq 1$. □

Lemma 2. *Under Condition 1, $V_i(s) \leq H_i(s)$, for any $s \in \Omega$.*

Lemma 2 is from [33]. Here, we borrow this lemma and skip its proof.

Proof of Condition 1: By Lemma 1, it's straightforward that

$$\mathbb{E}\left[h_i(S_{n,i})|S_{0,i} = s, i_1 =, \ldots, i_n = i\right] - h_i(s) \leq \frac{a\sqrt{2\pi}}{\sqrt{b}} G(d, \eta, b, \lambda) - h_i(s) = H_i(s),$$

for any $s = (\eta, \lambda) \in \Omega$.

Therefore, Condition 1 holds. □

Proof of Condition 2: Define $f(u) = \max\{\Phi(u), 1 - \Phi(u)\}$. Then

$$\begin{aligned}
&\mathbb{E}\left[h_i(S_{1,i})|S_{0,i} = s, i_1 = i\right] - h_i(s) \\
&= \mathbb{E}\left[h_i\left(\eta + \tilde{\sigma}_i(\lambda) \cdot Z, \lambda + \gamma_i\right)\right] - h_i(\eta, \lambda) \\
&= \frac{a\sqrt{2\pi}}{\sqrt{b}} \mathbb{E}\left[G\left(d, \eta + \tilde{\sigma}_i(\lambda) \cdot Z, b, \lambda + \gamma_i\right) \cdot f\left(\frac{\lambda + \gamma_i}{\sqrt{b + \lambda + \gamma_i}}(d - \eta - \tilde{\sigma}_i(\lambda) \cdot Z)\right)\right] \\
&\quad - h_i(\eta, \lambda) \\
&\leq \frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot \mathbb{E}\left[G\left(d, \eta + \tilde{\sigma}_i(\lambda) \cdot Z, b, \lambda + \gamma_i\right)\right] \cdot \mathbb{E}\left[f\left(\frac{\lambda + \gamma_i}{\sqrt{b + \lambda + \gamma_i}}(d - \eta - \tilde{\sigma}_i(\lambda) \cdot Z)\right)\right] \\
&\quad - h_i(\eta, \lambda) \\
&= \frac{a\sqrt{2\pi}}{\sqrt{b}} \cdot G(d, \eta, b, \lambda) \\
&\quad \cdot \left\{\mathbb{E}\left[f\left(\frac{\lambda + \gamma_i}{\sqrt{b + \lambda + \gamma_i}}(d - \eta - \tilde{\sigma}_i(\lambda) \cdot Z)\right)\right] - f\left(\frac{\lambda}{\sqrt{b + \lambda}}(d - \eta)\right)\right\} \\
&\leq \frac{a\sqrt{\lambda}}{\sqrt{b + \lambda}} \cdot \left\{\mathbb{E}\left[f\left(\frac{\lambda + \gamma_i}{\sqrt{b + \lambda + \gamma_i}}(d - \eta - \tilde{\sigma}_i(\lambda) \cdot Z)\right)\right] - f\left(\frac{\lambda}{\sqrt{b + \lambda}}(d - \eta)\right)\right\}.
\end{aligned}$$

From the e-companion of [33], we have that

$$\begin{aligned}
&\mathbb{E}\left[f\left(\frac{\lambda + \gamma_i}{\sqrt{b + \lambda + \gamma_i}}(d - \eta - \tilde{\sigma}_i(\lambda) \cdot Z)\right)\right] - f\left(\frac{\lambda}{\sqrt{b + \lambda}}(d - \eta)\right) \\
&\leq \left(\sqrt{1 + \beta_i^{\epsilon}/\beta} - 1\right)\left(1 + 1/\sqrt{2\pi e} + \pi^{-1}\sqrt{\beta_i^{\epsilon}/\beta}\right),
\end{aligned}$$

where $\beta = \frac{\lambda^2}{b+\lambda}$ and $\beta_i^\epsilon = \frac{(\lambda+\gamma_i)^2}{b+\lambda+\gamma_i} - \frac{\lambda^2}{b+\lambda}$.

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[h_i(S_{1,i})|S_{0,i} = s, i_1 = i\right] - h_i(s) \\
&\leq \frac{a\sqrt{\lambda}}{\sqrt{b+\lambda}} \cdot \left(\sqrt{1+\beta_i^\epsilon/\beta} - 1\right)\left(1 + 1/\sqrt{2\pi e} + \pi^{-1}\sqrt{\beta_i^\epsilon/\beta}\right) \\
&= \tilde{H}_i(\eta, \lambda).
\end{aligned}$$

Now, since $\lim_{n\to\infty} \beta_i^\epsilon/\beta = 0$,

$$\lim_{n\to\infty}\left[\sup_{s\in PS(i,;n)} \tilde{H}_i(s)\right] = 0,$$

where

$$PS(i;n) = \{s \in \Omega : \exists s' \in \Omega \text{ s.t. } \Pr\{S_{n,i} = s | S_{0,i} = s', i_1 = i_2 = \ldots = i_n = i\} > 0\}.$$

Therefore, Condition 2 holds. □

Proof of Condition 3: Pick any $\lambda > 0$. As $\eta \to \infty$, it is straightforward that $h_i(\eta, \lambda) \to 0$. Consequently, $R_i(\eta, \lambda) \to -c$ almost surely as $\eta \to \infty$.

In addition, by Lemma 2,

$$\mathbb{E}\left[V_i(\eta + \tilde{\sigma}_i(\lambda)\cdot Z, \lambda + \gamma_i)\right] \leq \mathbb{E}\left[H_i(\eta + \tilde{\sigma}_i(\lambda)\cdot Z, \lambda + \gamma_i)\right] \to 0,$$

almost surely as $\eta \to \infty$.

As a result, $\limsup_{\eta\to\infty} L_i(\eta, \lambda, V_i) < 0$ almost surely. We can therefore, find some $\overline{\eta_i}(\lambda)$ such that for all $\eta > \overline{\eta_i}(\lambda)$, $L_i(\eta, \lambda, V_i) < 0$ and hence $V_i(\eta, \lambda) = 0$. Similar arguments can prove there exists some $\underline{\eta_i}(\lambda)$ such that for all $\eta < \underline{\eta_i}(\lambda)$, $V_i(\eta, \lambda) = 0$. Thus, Condition 3 holds.

A.2 Derivation of Bayesian Optimal Policy for the Normal Reward Function

Define

$$R_i(\eta, \lambda) = \mathbb{E}\left[h_i(\eta + \tilde{\sigma}_i(\lambda) \cdot Z, \lambda + \gamma_i)\right] - h_i(\eta, \lambda) - c,$$

where $Z \sim \mathcal{N}(0, 1)$. We only need to find $\mathbb{E}\left[h(\eta + \tilde{\sigma}_i(\lambda) \cdot Z, \lambda + \gamma_i)\right]$.

We know that

$$\begin{aligned}
&\mathbb{E}\left[h(\eta + \tilde{\sigma}_i(\lambda) \cdot Z, \lambda + \gamma_i)\right] \\
&= \int_{-\infty}^{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}} h_0(\eta + \tilde{\sigma}_i(\lambda) \cdot z, \lambda + \gamma_i) \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2} z^2\} dz \qquad \text{(A.1)} \\
&\quad + \int_{\frac{d-\eta}{\tilde{\sigma}(\lambda)}}^{\infty} h_1(\eta + \tilde{\sigma}_i(\lambda) \cdot z, \lambda + \gamma_i) \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2} z^2\} dz. \qquad \text{(A.2)}
\end{aligned}$$

We can compute each part separately as follows:

$$
\begin{aligned}
(A.1) =& \frac{a\sqrt{2\pi}}{\sqrt{b}} \int_{-\infty}^{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}} G(d, \eta + \tilde{\sigma}_i(\lambda)\cdot z, b, \lambda+\gamma_i) \qquad (A.3)\\
& \cdot \Phi((d - \frac{db + (\eta+\tilde{\sigma}_i(\lambda)\cdot z)(\lambda+\gamma_i)}{b+\lambda+\gamma_i})\sqrt{b+\lambda+\gamma_i}) \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}z^2\} dz \\
=& \frac{a\sqrt{2\pi}}{\sqrt{b}} \int_{-\infty}^{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b(\lambda+\gamma_i)}{b+\lambda+\gamma_i}} \exp\{-\frac{1}{2}(d-(\eta+\tilde{\sigma}_i(\lambda)\cdot z))^2 \frac{b(\lambda+\gamma_i)}{b+\lambda+\gamma_i}\} \\
& \cdot \Phi((d - \frac{db + (\eta+\tilde{\sigma}_i(\lambda)\cdot z)(\lambda+\gamma_i)}{b+\lambda+\gamma_i})\sqrt{b+\lambda+\gamma_i}) \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}z^2\} dz \\
=& \frac{a\sqrt{2\pi}}{\sqrt{b}} \sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} \int_{-\infty}^{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}} \\
& \cdot \exp\{-\frac{1}{2}(z - \frac{d-\eta}{\tilde{\sigma}_i(\lambda)})^2 \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}\} \\
& \cdot \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}z^2\} \Phi((d - \frac{db + (\eta+\tilde{\sigma}_i(\lambda)\cdot z)(\lambda+\gamma_i)}{b+\lambda+\gamma_i})\sqrt{b+\lambda+\gamma_i}) dz \\
=& \frac{a\sqrt{2\pi}}{\sqrt{b}} \sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} G(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, 0, \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}, 1) \int_{-\infty}^{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}} \frac{1}{\sqrt{2\pi}} \sqrt{\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)} + 1} \\
& \cdot \exp\{-\frac{1}{2}(z - \frac{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)} b\gamma_i}{b\gamma_i + \lambda(b+\lambda+\gamma_i)})^2 \cdot (\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)} + 1)\} \\
& \cdot \Phi((d - \frac{db + (\eta+\tilde{\sigma}_i(\lambda)\cdot z)(\lambda+\gamma_i)}{b+\lambda+\gamma_i})\sqrt{b+\lambda+\gamma_i}) dz \\
=& \frac{a\sqrt{2\pi}}{\sqrt{b}} \sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}} G(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, 0, \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}, 1) \\
& \cdot Pr\{Z_1 \le \frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, Z_2 \le (d - \frac{db + (\eta+\tilde{\sigma}_i(\lambda)\cdot Z_1)(\lambda+\gamma_i)}{b+\lambda+\gamma_i})\sqrt{b+\lambda+\gamma_i}\}, \qquad (A.4)
\end{aligned}
$$

where

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \sim MVN(\mathbf{m}, \mathbf{\Sigma}),$$

and

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} \frac{\frac{d-\eta}{\tilde{\sigma}_i(\lambda)} b\gamma_i}{b\gamma_i + \lambda(b+\lambda+\gamma_i)} \\ 0 \end{bmatrix},$$

$$\mathbf{\Sigma} = \begin{bmatrix} 1/b_1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1/(\frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)} + 1) & 0 \\ 0 & 1 \end{bmatrix}.$$

Let $Z_3 = Z_2 - (d - \frac{db+(\eta+\tilde{\sigma}_i(\lambda)\cdot Z_1)(\lambda+\gamma_i)}{b+\lambda+\gamma_i})\sqrt{b+\lambda+\gamma_i}$. Then

$$\text{(A.4)} = \Pr\{Z_1 \leq \frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, Z_3 \leq 0\}.$$

Moreover,

$$\mathbb{E}[Z_3] = -\frac{(d-\eta)(\lambda+\gamma_i)}{\sqrt{b+\lambda+\gamma_i}} + \frac{\tilde{\sigma}_i(\lambda)(\lambda+\gamma_i)}{\sqrt{b+\lambda+\gamma_i}} m_1 = -A + Bm_1;$$

$$\text{Var}[Z_3] = 1 + B^2/b_1;$$

$$\begin{aligned} \text{Cov}[Z_1, Z_3] =& \mathbb{E}[Z_1 Z_3] - \mathbb{E}[Z_1]E[Z_3] \\ =& - Am_1 + B \cdot \mathbb{E}[Z_1^2] - m_1 \cdot (-A + Bm_1) \\ =& B/b_1. \end{aligned}$$

Therefore,

$$\begin{bmatrix} Z_1 \\ Z_3 \end{bmatrix} \sim MVN(\tilde{\mathbf{m}}, \tilde{\mathbf{\Sigma}}),$$

where

$$\tilde{\mathbf{m}} = \begin{bmatrix} m_1 \\ m_3 \end{bmatrix},$$

$$\tilde{\mathbf{\Sigma}} = \begin{bmatrix} 1/b_1 & B/b_1 \\ B/b_1 & 1 + B^2/b_1. \end{bmatrix}$$

Finally,

$$\text{(A.1)} = \frac{a\sqrt{2\pi}}{\sqrt{b}}\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}}G(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, 0, \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}, 1)\cdot \Pr\{Z_1 \le \frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, Z_3 \le 0\},$$

which can be calculated by using the cumulative distribution function of a bivariate normal vector.

Using similar arguments,

$$\text{(A.2)} = \frac{a\sqrt{2\pi}}{\sqrt{b}}\sqrt{\frac{\lambda(\lambda+\gamma_i)}{\gamma_i}}G(\frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, 0, \frac{b\gamma_i}{\lambda(b+\lambda+\gamma_i)}, 1)\cdot \Pr\{Z_1 \ge \frac{d-\eta}{\tilde{\sigma}_i(\lambda)}, Z_3 \ge 0\}.$$

A.3 The $\mathcal{BK}$ Procedure

The $\mathcal{BK}$ procedure employs the concept of tolerance level ϵ, which is the amount away from the constraint threshold d that the decision maker is willing to take. Given ϵ and d, system i is

- desirable if $\mu_i \le d - \epsilon$;
- undesirable if $\mu_i \ge d + \epsilon$; and
- acceptable if $d - \epsilon < \mu_i < d + \epsilon$.

Note that system i is defined as feasible if $\mu_i \le d$. In general, with a finite number of observations, it is impossible to detect the feasibility of system i with 100% guarantee, especially when μ_i is close to d or equal to d. Instead, the $\mathcal{BK}$ procedure tries to provide a probability guarantee of finding near-feasible systems.

Let $K = \{1, 2, \ldots, k\}$ (the set of all systems), $F_D = \{i : \mu_i \le d-\epsilon\}$ (the set of desirable systems), $F_U = \{i : \mu_i \ge d+\epsilon\}$ (the set of undesirable systems) and $F_A = K \setminus (F_D \cup F_U)$

(the set of acceptable systems). Then $\text{PCD}_{\mathcal{BK}}$ is defined as follows:

$$\text{PCD}_{\mathcal{BK}} = \Pr\{F_D \subset F \subset (F_D \cup F_A)\} \geq 1 - \alpha, \tag{A.5}$$

where F is the set of systems returned by the $\mathcal{BK}$ procedure and $1 - \alpha$ is the nominal confidence level.

The main statistics of the $\mathcal{BK}$ procedure are $\sum_{j=1}^{n}(Y_{ij} - d)$, the cumulative sums of differences between n observations and the constraint threshold d. Whenever a new observation is taken, the partial sum is updated and it is checked whether the partial sum exits, so called, a continuation region. If the partial sum stays within the continuation region, then sampling continues. Otherwise, a decision on the feasibility of the system is made. In a sense that only one observation is obtained between checking the stopping rule, this procedure is called fully sequential. The continuation region is constructed so that $\text{PCD}_{\mathcal{BK}}$ is at least a pre-specified confidence level $(1-\alpha)$. Under single constraint, the $\mathcal{BK}$ procedure is identical with the feasibility determination procedure in [26], and [26] proves that the $\mathcal{BK}$ procedure satisfies (A.5). The $\mathcal{BK}$ procedure is stated in Algorithm 6.

A.4 The $\mathcal{GC}$ Procedure

The $\mathcal{GC}$ procedure employs the large deviation approach for the feasibility determination problem under general distribution assumption. It aims to maximize PCD, given a simulation budget $\mathcal{B}$. Since we set that $\mu_1, \ldots, \mu_{25} \leq d$ and $\mu_{26}, \ldots, \mu_{50} > d$. Then the problem is formulated as

$$\begin{aligned} &\max_{p_1, p_2, \ldots, p_k} \text{PCD} \\ &\text{s.t.} \sum_{i=1}^{50} p_i = 1, \end{aligned} \tag{A.6}$$

Algorithm 6 $\mathcal{BK}$ procedure

1: **Setup**: Select first-stage sample size $n_0 \geq 2$ and nominal confidence level $0 \leq 1-\alpha < 1$. For the constraint under consideration, choose tolerance level ϵ and d. Compute ξ as the solution to

$$\xi = -\log\left\{2(1-(1-\alpha)^{1/k})\right\}.$$

2: **Initialization**: Let $M = \{1, 2, \ldots, k\}$ and $F = \emptyset$ be the set of systems whose feasibility is not determined yet and the set of set of feasible systems, respectively. Obtain n_0 observations Y_{ij}, $j = 1, 2, \ldots, n_0$, from each system i. Set the stage counter $n = n_0$ and go to **Feasibility Check**.

3: **Feasibility Check**: For each system $i \in M$, calculate

$$R(n) = \max\left(0, \frac{\xi}{\epsilon\gamma_i} - \frac{\epsilon}{2}n\right).$$

If $\sum_{j=1}^{n}(Y_{ij} - d) \leq -R(n)$, move i from M to F; else if $\sum_{j=1}^{t}(Y_{ij} - d) \geq R(n)$, then eliminate i from M.

4: **Termination Rule**: If $\|M\| = 0$, then return F as a set of feasible systems. Otherwise, take one additional observation $Y_{i,n+1}$ from each system $i \in M$, set $n = n + 1$, and go to **Feasibility Check**.

where

$$\text{PCD} = \Pr((\cap_{i=1}^{25}\bar{Y}_i \leq d) \cup (\cap_{i=26}^{50}\bar{Y}_i > d)).$$

Note that p_i's represent the fraction of the simulation budget that is allocated to sampling from each system, and $\bar{Y}_i$ is the sample mean after allocating all N budgets.

A major difficulty associated with (A.6) is that a convenient closed-form expression for PCD is unknown, other than a few special cases. However, by using an approximate PCD (APCD), which is based on the large deviation rate functions, [32] is able to provide an asymptotically optimal budget allocation rule that maximizes APCD. In addition, the large deviation rate functions, denoted as I_i for each system i, have explicit forms for calculation so that they don't need to be estimated. The $\mathcal{GC}$ procedure is then constructed as in Algorithm 7.

Algorithm 7 $\mathcal{GC}$ procedure

1: **Setup**: Specify total simulation budget $\mathcal{B}$, initial simulation replication number n_0, incremental budget Δ_0, and threshold d.

2: **Initialization**: For each design i, obtain n_0 observations Y_{ij}, $j = 1, 2, \ldots, n_0$. Compute $\bar{Y}_i$, the sample mean, and set $n_i = n_0$. Compute

$$I_i = \frac{1}{2}(\bar{Y}_i - d)^2 \gamma_i \cdot n_i.$$

Set $p_i = 1/k$ for $i = 1, 2, \ldots, k$.

3: **Iteration**: If $\sum_{i=1}^{k} n_i = \mathcal{B}$, stop and go to **Decision**. Otherwise, provide an incremental simulation budget $\Delta = \min\left\{\Delta_0, N - \sum_{i=1}^{k} n_i^r\right\}$ and allocate it to the k designs according to proportions $I_i^{-1} / \sum_{s=1}^{k} I_s^{-1}$, for $i = 1, .., k$. Let Δ_i be the number of additional replications to design i. Then set $n_i = n_i + \Delta_i$. Update p_i, $\bar{Y}_i$, and I_i, for $i = 1, \ldots, k$.

4: **Decision**: Return $F = \left\{i : \bar{Y}_i < d\right\}$ as the feasible set.

APPENDIX B

SUPPORTING MATERIALS FOR CHAPTER IV

In this appendix, we show the proof of Theorem 2, which is based on the following assumptions:

- The prior mean δ_0 and covariance Σ_0 and sampling variances $\sigma_s^2(\mathbf{x})$'s are known and fixed;
- The covariance Σ_0 is positive definite.

We consider both stochastic simulation (i.e., $\sigma_s^2(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbf{X}$) and deterministic simulation (i.e., $\sigma_s^2(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{X}$). The following lemmas assume that $\sigma_s^2(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbf{X}$ and will be used for the proof of Theorem 2 in the case of stochastic simulation.

First, we introduce some useful functions defined as follows:

- $f(u) = \max\{\Phi(u), 1 - \Phi(u)\}$,
- $\alpha(u) = \begin{cases} \sup_v \frac{\Phi(-v)-\Phi(u)}{u-v} & \text{if } u \geq 1/2, \\ \sup_v \frac{\Phi(v)-\Phi(-u)}{v-u} & \text{if } u < 1/2, \end{cases}$
- $\beta(u) = \begin{cases} \sup_{v \geq u} \frac{\Phi(v)-\Phi(u)}{v-u} & \text{if } u \geq 1/2, \\ \sup_{v \leq u} \frac{\Phi(v)-\Phi(u)}{v-u} & \text{if } u < 1/2, \end{cases}$
- $s(u) = \max\{\alpha(u), \beta(u)\}$,
- $g(u, v) = f(u) + |v - u|s(u)$,

where $u, v \in \mathbb{R}$.

Next, we state the following lemmas. The proofs of Lemmas 3, 4 and 5 can be found in [24], while the proof of Lemma 6 can be found in the supplemental material of [33].

Lemma 3. *There exist random variables δ_∞, Σ_∞ such that δ_n and Σ_n respectively converge to δ_∞ and Σ_∞ almost surely.*

Lemma 4. *For all $\mathbf{x} \in \mathbf{X}$ and $n \in \mathbb{Z}$, $\Sigma_{n+1}(\mathbf{x}) \leq \Sigma_n(\mathbf{x})$.*

Lemma 5. *If alternative $\mathbf{x}$ is sampled infinitely often, then $\lim_{n\to\infty} \Sigma_n(\mathbf{x}) = 0$ almost surely.*

Lemma 6. *The functions $f(\cdot)$, $s(\cdot)$ and $g(\cdot,\cdot)$ satisfy the following:*

- $0 \leq s(u) \leq 1/\sqrt{2\pi}$,
- $|u|s(u) \leq 1 + 1/\sqrt{2\pi e}$,
- $f(v) \leq g(u, v)$,

for all $u, v \in \mathbb{R}$.

Then, we state a few more lemmas and their proofs.

Lemma 7. *If alternative $\mathbf{x}$ is sampled infinitely often, then $\lim_{n\to\infty}|\Sigma_n(\mathbf{x}, \mathbf{x}')| = 0$ almost surely.*

Proof. By using the Cauchy-Schwart inequality,

$$|\Sigma_n(\mathbf{x}, \mathbf{x}')|^2 \leq \Sigma_n(\mathbf{x}) \cdot \Sigma_n(\mathbf{x}') \implies |\Sigma_n(\mathbf{x}, \mathbf{x}')| \leq \sqrt{\Sigma_n(\mathbf{x}) \cdot \Sigma_n(\mathbf{x}')}.$$

Since $\Sigma_n(\mathbf{x}')$ is bounded below by 0 and monotonically decreasing by Lemma 4, we know $\lim_{n\to\infty} \Sigma_n(\mathbf{x}')$ exists and is less than ∞. By Lemma 5, if alternative $\mathbf{x}$ is sampled infinitely often, $\lim_{n\to\infty} \Sigma_n(\mathbf{x}) = 0$. Therefore, $\lim_{n\to\infty}|\Sigma_n(\mathbf{x}, \mathbf{x}')| = 0$ almost surely. □

Lemma 8. *For any* $\mathbf{x}, \mathbf{x}' \in \mathbf{X}$, $\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] - \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right] > 0$. *Consequently,* $\text{VOI}_n(\mathbf{x}) > 0$ *for any* $\mathbf{x} \in \mathbf{X}$ *and any finite positive integer* n.

Proof. Define $P_n(\mathbf{x}') = \Phi\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right)$ and $\tilde{f}(p) = \max\{p, 1-p\}$ on $[0,1]$. By Tower rule, we can show

$$\begin{aligned}
\mathbb{E}_n\left[P_{n+1}(\mathbf{x}')|\mathbf{x}_{n+1} = \mathbf{x}\right] &= \mathbb{E}_n\left[\mathbb{E}_{n+1}\left[I\left(\mu(\mathbf{x}') \leq d\right)|\mathbf{x}_{n+1} = \mathbf{x}\right]|\mathbf{x}_{n+1} = \mathbf{x}\right] \\
&= \mathbb{E}_n\left[I\left(\mu(\mathbf{x}') \leq d\right)|\mathbf{x}_{n+1} = \mathbf{x}\right] \\
&= P_n(\mathbf{x}').
\end{aligned}$$

Since $\tilde{f}$ is a convex function, by Jensen's inequality, we have

$$\begin{aligned}
\mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right] &= \tilde{f}\left(P_n(\mathbf{x}')\right) \\
&= \tilde{f}\left(\mathbb{E}_n\left[P_{n+1}(\mathbf{x}')|\mathbf{x}_{n+1} = \mathbf{x}\right]\right) \\
&< \mathbb{E}_n\left[\tilde{f}\left(P_{n+1}(\mathbf{x}')\right)|\mathbf{x}_{n+1} = \mathbf{x}\right] \\
&= \mathbb{E}_n\left[\mathbb{E}_{n+1}\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)\right]|\mathbf{x}_{n+1} = \mathbf{x}\right] \\
&= \mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right].
\end{aligned}$$

By the definition of $\text{VOI}_n(\mathbf{x})$ as in (4.3), we have $\text{VOI}_n(\mathbf{x}) > 0$. □

Lemma 9. *If alternative* $\mathbf{x}$ *is sampled infinitely often, then* $\lim_{n\to\infty} \text{VOI}_n(\mathbf{x}) = 0$ *almost surely.*

Proof. Let $\mathbf{x}'$ be an arbitrary element in $\mathbf{X}$. We start the proof by finding an upper bound of $\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] - \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right]$. Define $\tilde{\sigma}_n(\mathbf{x}') = \frac{\Sigma_n(\mathbf{x}', \mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x}) + \sigma_s^2(\mathbf{x})}}$. By Lemma 5 and Lemma 7, we know $\lim_{n\to\infty} |\tilde{\sigma}_n(\mathbf{x}')| = 0$ almost surely.

Since, by Lemma 4, $\Sigma_n(\mathbf{x}')$ is monotonically decreasing and bounded below by zero, we know that $\lim_{n\to\infty} \Sigma_n(\mathbf{x}')$ exists.

First suppose that $\lim_{n\to\infty} \Sigma_n(\mathbf{x}') = 0$ almost surely.

$$\lim_{n\to\infty} \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right] = \lim_{n\to\infty} \max\left\{\Phi\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right), 1-\Phi\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right)\right\}$$
$$= 1, \text{ almost surely.}$$

Since $1 \geq \mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] \geq \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right]$ for all $n \in \mathbb{Z}^+$, we know that

$$\lim_{n\to\infty} \mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] = 1 \text{ almost surely.}$$

Therefore, $\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] - \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right] \to 0$ as $n \to \infty$ almost surely.

Now suppose $\lim_{n\to\infty} \Sigma_n(\mathbf{x}') > 0$ almost surely. Then,

$$\begin{aligned}
&\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}'; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] - \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}'; \mu, d)\right] \\
&= \mathbb{E}_n\left[f\left(\frac{d-\delta_{n+1}(\mathbf{x}')}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right)\right] - f\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&= \mathbb{E}_n\left[f\left(\frac{d-\delta_n(\mathbf{x}')-\tilde{\sigma}_n(\mathbf{x}')Z}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right)\right] - f\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&\leq \mathbb{E}_n\left[g\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}, \frac{d-\delta_n(\mathbf{x}')-\tilde{\sigma}_n(\mathbf{x}')Z}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right)\right] - f\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&= \mathbb{E}_n\left[\left|\frac{d-\delta_n(\mathbf{x}')-\tilde{\sigma}_n(\mathbf{x}')Z}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} - \frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right|\right] \cdot s\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&= \mathbb{E}_n\left[\left|\left(\frac{1}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} - \frac{1}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right)(d-\delta_n(\mathbf{x}')) - \frac{\tilde{\sigma}_n(\mathbf{x})'}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} \cdot Z\right|\right] \cdot s\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&\leq \left(\frac{1}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} - \frac{1}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}}\right)|d-\delta_n(\mathbf{x}')| + \frac{|\tilde{\sigma}_n(\mathbf{x}')|}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} \cdot \mathbb{E}_n\left[|Z|\right] \cdot s\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&= \left(\frac{\sqrt{\Sigma_n(\mathbf{x}')}}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} - 1\right)\left|\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right| \cdot s\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) + \frac{|\tilde{\sigma}_n(\mathbf{x}')|}{\sqrt{\Sigma_{n+1}}(\mathbf{x}')}\sqrt{\frac{2}{\pi}} \cdot s\left(\frac{d-\delta_n(\mathbf{x}')}{\sqrt{\Sigma_n(\mathbf{x}')}}\right) \\
&\leq \left(\frac{\sqrt{\Sigma_n(\mathbf{x}')}}{\sqrt{\Sigma_{n+1}(\mathbf{x}')}} - 1\right)\left(1 + \frac{1}{\sqrt{2\pi e}}\right) + \frac{|\tilde{\sigma}_n(\mathbf{x}')|}{\sqrt{\Sigma_{n+1}(\mathbf{x}')\pi}} \to 0 \text{ almost surely as } n \to \infty.
\end{aligned}$$

Therefore, $\lim_{n\to\infty} \text{VOI}_n(\mathbf{x}) = 0$ almost surely. □

Lemma 10. *If* $\liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) = 0$ *almost surely for all* $\mathbf{x} \in \mathbf{X}$*, then* $\lim_{n\to\infty} \Sigma_n(\mathbf{x}) = 0$ *almost surely for all* $\mathbf{x} \in \mathbf{X}$.

Proof. We prove the contrapositive statement of this lemma.

Suppose $\max_{\mathbf{x}\in\mathbf{X}} \{\lim_{n\to\infty} \Sigma_n(\mathbf{x})\} > 0$. Then there exists $\mathbf{x} \in \mathbf{X}$ such that $\lim_{n\to\infty} \Sigma_n(\mathbf{x}) = \epsilon(\mathbf{x}) > 0$, and this implies that $\mathbf{x}$ is not sampled infinitely often as $n \to \infty$. As $n \to \infty$, at least one $\mathbf{x}' \in \mathbf{X}$ is sampled infinitely often.

From Lemma 3, we have that $\delta_n(\mathbf{x}) \to \delta_\infty(\mathbf{x})$ almost surely as $n \to \infty$. In addition, define $\tilde{\sigma}_n(\mathbf{x}) = \frac{\Sigma_n(\mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x})+\sigma_s^2(\mathbf{x})}}$, then $\lim_{n\to\infty} \tilde{\sigma}_n(\mathbf{x}) = \frac{\epsilon(\mathbf{x})}{\sqrt{\epsilon(\mathbf{x})+\sigma_s^2(\mathbf{x})}}$ almost surely.

Since $\text{VOI}_n(\mathbf{x}) > \mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] - \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}; \mu, d)\right] > 0$ by Lemma 8,

$$
\begin{aligned}
&\liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) \\
&\geq \lim_{n\to\infty} \left\{\mathbb{E}_n\left[r(\mathbf{F}_{n+1}, \mathbf{x}; \mu, d)|\mathbf{x}_{n+1} = \mathbf{x}\right] - \mathbb{E}_n\left[r(\mathbf{F}_n, \mathbf{x}; \mu, d)\right]\right\} \\
&= \lim_{n\to\infty} \left\{\mathbb{E}\left[f\left(\Phi\left(\frac{d - \delta_n(\mathbf{x}) - \tilde{\sigma}_n(\mathbf{x}) \cdot Z}{\sqrt{\Sigma_{n+1}(\mathbf{x})}}\right)\right)\right] - f\left(\Phi\left(\frac{d - \delta_n(\mathbf{x})}{\sqrt{\Sigma_n(\mathbf{x})}}\right)\right)\right\} \\
&= \mathbb{E}\left[f\left(\Phi\left(\frac{d - \delta_\infty(\mathbf{x}) - \frac{\epsilon(\mathbf{x})}{\sqrt{\epsilon(\mathbf{x})+\sigma_s^2(\mathbf{x})}} \cdot Z}{\sqrt{\epsilon(\mathbf{x})}}\right)\right)\right] - f\left(\Phi\left(\frac{d - \delta_\infty(\mathbf{x})}{\sqrt{\epsilon(\mathbf{x})}}\right)\right).
\end{aligned}
$$

Since, for any possible value of $\delta_\infty(\mathbf{x})$, we can show that

$$
\mathbb{E}\left[f\left(\Phi\left(\frac{d - \delta_\infty(\mathbf{x}) - \frac{\epsilon(\mathbf{x})}{\sqrt{\epsilon(\mathbf{x})+\sigma_s^2(\mathbf{x})}} \cdot Z}{\sqrt{\epsilon(\mathbf{x})}}\right)\right)\right] - f\left(\Phi\left(\frac{d - \delta_\infty(\mathbf{x})}{\sqrt{\epsilon(\mathbf{x})}}\right)\right) > 0
$$

by the Jensen's inequality. Therefore, $\liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) > 0$ almost surely. □

Finally, we present the proof of Theorem 2.

Proof of Theorem 2: We divide the proof into two cases. First, we consider stochastic simulation where $\sigma_s^2(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbf{X}$. We first show, by contradiction, that

$\liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) = 0$ almost surely for all $\mathbf{x} \in \mathbf{X}$. Consider an arbitrary sample path taken by the $\mathcal{GPFD}$ procedure. Let

$$\tilde{\mathbf{X}}_0 = \left\{\mathbf{x} \in \mathbf{X} : \lim_{n\to\infty} \text{VOI}_n(\mathbf{x}) \text{ exists and is } 0\right\};$$
$$\tilde{\mathbf{X}}_1 = \left\{\mathbf{x} \in \mathbf{X} : \liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) = 0\right\}.$$

Suppose that $\tilde{\mathbf{X}}_1 \neq \mathbf{X}$, which implies $\mathbf{X} \setminus \tilde{\mathbf{X}}_1 = \{\mathbf{x} \in \mathbf{X} : \liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) > 0\}$ is not empty. Pick $\mathbf{x} \in \mathbf{X}\setminus\tilde{\mathbf{X}}_1$. Then there exist some $\xi > 0$ and a subsequence of $\mathbb{Z}^+$, denoted as $\{n_j\}_{j=1}^{\infty}$ such that $\text{VOI}_{n_j}(\mathbf{x}) \geq \xi$ for all $j = 1, 2, \ldots$. Then we have $\text{VOI}_{n_j}(\mathbf{x}_{n_j}) \geq \xi$ for all $j = 1, 2, \ldots$, where $\mathbf{x}_{n_j}$ is the alternative chosen to sample at stage n_j.

For each $\mathbf{x}' \in \mathbf{X} \setminus \tilde{\mathbf{X}}_0$, the contrapositive of Lemma 9 implies that there exists a finite number $\tilde{N}(\mathbf{x}')$ such that the $\mathcal{GPFD}$ procedure does not choose $\mathbf{x}'$ for $n > \tilde{N}(\mathbf{x}')$. Let $\tilde{N}^* = \max_{\mathbf{x}'\in\mathbf{X}\setminus\tilde{\mathbf{X}}_0} \tilde{N}(\mathbf{x}')$. Therefore, $\mathbf{x}_n \in \tilde{\mathbf{X}}_0$ for all $n > \tilde{N}^*$.

For each $\mathbf{x}' \in \tilde{\mathbf{X}}_0$, $\lim_{n\to\infty} \text{VOI}_n(\mathbf{x}') = 0$. Therefore, there exists a finite number $\tilde{N}_0(\mathbf{x}')$ such that $\text{VOI}_n(\mathbf{x}') < \xi$ for all $n > \tilde{N}_0(\mathbf{x}')$. Let $\tilde{N}_0^* = \max_{\mathbf{x}'\in\tilde{\mathbf{X}}_0} \tilde{N}_0(\mathbf{x}')$. Then for all $n > \tilde{N}_0^*$, $\text{VOI}_n(\mathbf{x}') < \xi$ for any $\mathbf{x}' \in \tilde{\mathbf{X}}_0$.

It follows that for all $n > \max\left\{\tilde{N}^*, \tilde{N}_0^*\right\}$, $\text{VOI}_n(\mathbf{x}_n) < \xi$, which contradicts that $\text{VOI}_{n_j}(\mathbf{x}_{n_j}) \geq \xi$ for all j. We thus conclude that $\tilde{\mathbf{X}}_1 = \mathbf{X}$. Since the sample path considered is arbitrary, we conclude that $\liminf_{n\to\infty} \text{VOI}_n(\mathbf{x}) = 0$ almost surely for all $\mathbf{x} \in \mathbf{X}$.

By Lemma 10, we have that $\lim_{n\to\infty} \Sigma_n(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{X}$ almost surely. By Lemma 3, we have that δ_n converges to δ_∞ almost surely. Therefore,

$$\begin{aligned}\lim_{n\to\infty} \mathbb{E}_n\left[R(\mathbf{F};\mu,d)\right] &= \sum_{\mathbf{x}\in\mathbf{X}} \max\left\{\Phi\left(\frac{d-\delta_n(\mathbf{x})}{\Sigma_n(\mathbf{x})}\right), 1-\Phi\left(\frac{d-\delta_n(\mathbf{x})}{\Sigma_n(\mathbf{x})}\right)\right\}\\ &= |\mathbf{X}| \text{ almost surely,}\end{aligned}$$

where $|\mathbf{X}|$ is the cardinality of $\mathbf{X}$.

Now we consider deterministic simulation where $\sigma_s^2(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{X}$. For

$n = 1, 2, \ldots$, let the set of alternatives that have been sampled up to stage n be $\mathcal{X}_n = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ For any $\mathbf{x} \in \mathcal{X}_n$, we have that $\Sigma_n(\mathbf{x}) = 0$ and consequently, $\text{VOI}_n(\mathbf{x}) = 0$. On the other hand, for any $\mathbf{x} \notin \mathcal{X}_n$, $\Sigma_n(\mathbf{x}) > 0$, which leads to $\text{VOI}_n(\mathbf{x}) > 0$. As a result, at each stage, the $\mathcal{GPFD}$ chooses only the alternatives that have not been sampled before and all alternatives will be sampled for once only. After $n \geq |\mathbf{X}|$ stages, $\Sigma_n(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbf{X}$ and therefore, $\mathbb{E}_n\left[R(\mathbf{F}; \mu, d)\right] = |\mathbf{X}|$. □

APPENDIX C

SUPPORTING MATERIALS FOR CHAPTER V

In the Appendix, we present details of the mathematical models that we use in Phase I and II of our scheduling procedure.

C.1 The Phase I model M_I

The model M_I is given in Figure C.1. The first constraint in M_I ensures that each block is assigned only one bay with only one start date or outsourced. The second constraint ensures that for each bay, the sum of areas of assigned blocks to the bay does not exceed its available area over the entire planning horizon. The third constraint ensures the weights of assigned blocks in each bay does not exceed its crane capacity. The fourth constraint ensures that if a bay is not an eligible bay for a block, then the block is never assigned to the bay. The fifth constraint calculates space utilization $U(b,t)$ and the last constraint ensures that $\tilde{U}$ is the maximum of $U(b,t)$.

$$
\begin{aligned}
\min \quad & \textstyle\sum_{k\in \boldsymbol{K}} \Big[1 - \sum_{s\in \boldsymbol{ST}(k)} \sum_{b\in \boldsymbol{B}} d(k,s,b)\Big] + \tilde{U}, \\
\text{s.t.} \quad & \textstyle\sum_{s\in \boldsymbol{ST}(k)} \sum_{b\in \boldsymbol{B}} d(k,s,b) \le 1 \quad \forall k \in \boldsymbol{K}, \\
& \textstyle\sum_{k\in \boldsymbol{K}} \sum_{s\in \boldsymbol{ST}(k)} d(k,s,b) \times P(k,s,t) \times A(k) \le \text{TA}(b,t) \quad \forall b \in \boldsymbol{B},\ \forall t \in \boldsymbol{T}, \\
& d(k,s,b) \times V(k) \le C(b) \quad \forall k \in \boldsymbol{K},\ \forall s \in \boldsymbol{ST}(k),\ \forall b \in \boldsymbol{B}, \\
& d(k,s,b) \le I(k,b) \quad \forall k \in \boldsymbol{K},\ \forall s \in \boldsymbol{ST}(k),\ \forall b \in \boldsymbol{B}, \\
& d(k,s,b) = \tilde{d}(k,s,b) \quad \forall k \in \boldsymbol{PA},\ \forall s \in \boldsymbol{ST}(k),\ \forall b \in \boldsymbol{B}, \\
& \textstyle\sum_{k\in \boldsymbol{K}} \sum_{s\in \boldsymbol{ST}(k)} d(k,s,b) \times P(k,s,t) \times A(k) = \text{TA}(b,t) \times U(b,t) \\
& \quad \forall b \in \boldsymbol{B},\ \forall t \in \boldsymbol{T}, \\
& U(b,t) \le \tilde{U} \quad \forall b \in \boldsymbol{B},\ \forall t \in \boldsymbol{T}.
\end{aligned}
$$

Figure C.1: Mixed integer programming model for Phase I, M_I

C.2 The Phase II model M_{II}

The model M_{II} is given in Figure C.2. In M_{II}, the first constraint ensures that each block has exactly one coordinate assignment and one rotation assignment. The second and third constraints make the placement decisions of the already existing blocks consistent with their actual placement information. The remaining constraints make sure that none of the blocks are overlapping with the other.

$$
\begin{aligned}
\max \quad & \textstyle\sum_{k \in \boldsymbol{NA}(b,t)} \Big[\sum_{(v,h) \in \boldsymbol{SC}(k)} w(k,v,h) \times a_r(k,v,h) + \\
& \textstyle\sum_{(v,h) \in \boldsymbol{SC}_r(k)} w_r(k,v,h) \times a_r(k,v,h) \Big], \\
\text{s.t.} \quad & \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k)} a(k,v,h) + \sum_{(v,h) \in \boldsymbol{SC}_r(k)} a_r(k,v,h) = 1 \quad \forall k \in \boldsymbol{NA}(b,t), \\
& a(k,v,h) = \tilde{a}(k,v,h) \quad \forall k \in \boldsymbol{PA}(b,t), \\
& a_r(k,v,h) = \tilde{a}_r(k,v,h) \quad \forall k \in \boldsymbol{PA}(b,t), \\
& \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_2)} h \cdot a(k_2,v,h) + \sum_{(v,h) \in \boldsymbol{SC}(k_2)} h \cdot a_r(k_2,v,h) \\
& \quad - \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_1)} h \cdot a(k_1,v,h) - \sum_{(v,h) \in \boldsymbol{SC}(k_1)} h \cdot a_r(k_1,v,h) r \\
& \quad \geq W(k_1) \cdot hc(k_1,k_2) - M \cdot [1 - hc(k_1,k_2)] \\
& \quad \forall k_1, k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2, \\
& \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_2)} h \cdot a(k_2,v,h) + \sum_{(v,h) \in \boldsymbol{SC}(k_2)} h \cdot a_r(k_2,v,h) \\
& \quad - \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_1)} h \cdot a(k_1,v,h) - \sum_{(v,h) \in \boldsymbol{SC}(k_1)} h \cdot a_r(k_1,v,h) \\
& \quad \geq L(k_1) \cdot hc_r(k_1,k_2) - M \cdot [1 - hc_r(k_1,k_2)] \\
& \quad \forall k_1, k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2, \\
& \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_2)} v \cdot a(k_2,v,h) + \sum_{(v,h) \in \boldsymbol{SC}(k_2)} v \cdot a_r(k_2,v,h) \\
& \quad - \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_1)} v \cdot a(k_1,v,h) - \sum_{(v,h) \in \boldsymbol{SC}(k_1)} v \cdot a_r(k_1,v,h) \\
& \quad \geq L(k_1) \cdot vc(k_1,k_2) - M \cdot [1 - vc(k_1,k_2)] \\
& \quad \forall k_1, k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2, \\
& \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_2)} v \cdot a(k_2,v,h) + \sum_{(v,h) \in \boldsymbol{SC}(k_2)} v \cdot a_r(k_2,v,h) \\
& \quad - \textstyle\sum_{(v,h) \in \boldsymbol{SC}(k_1)} v \cdot a(k_1,v,h) - \sum_{(v,h) \in \boldsymbol{SC}(k_1)} v \cdot a_r(k_1,v,h) \\
& \quad \geq W(k_1) \cdot vc_r(k_1,k_2) - M \cdot [1 - vc_r(k_1,k_2)] \\
& \quad \forall k_1, k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2, \\
& \textstyle\sum_{k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2} [hc(k_1,k_2) + vc(k_1,k_2)] \leq M \cdot \sum \sum a(k_1,v,h), \\
& \textstyle\sum_{k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2} [hc_r(k_1,k_2) + vc_r(k_1,k_2)] \leq M \cdot \sum \sum a_r(k_1,v,h), \\
& hc(k_1,k_2) + hc_r(k_1,k_2) + vc(k_1,k_2) + vc_r(k_1,k_2) \geq 1 \\
& \quad \forall k_1, k_2 \in \boldsymbol{SB}(b,t), k_1 \neq k_2.
\end{aligned}
$$

Figure C.2: Integer programming model for Phase II, M_{II}

REFERENCES

[1] D. Goldsman and B. L. Nelson, "Comparing systems via simulation", in *Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice*, J. Banks, Ed. John Wiley & Sons, 1998, ch. 8, pp. 273–306.

[2] S.-H. Kim and B. L. Nelson, "Selecting the best system", in *Simulation*, S. G. Henderson and B. L. Nelson, Eds., ser. Handbooks in Operations Research and Management Science. Elsevier, 2006, vol. 13, ch. 17, pp. 501–534.

[3] A. M. Law and W. D. Kelton, *Simulation modeling and analysis*, 3rd ed., ser. McGraw-Hill series in industrial engineering and management science. McGraw-Hill, 2003.

[4] L. J. Hong, B. L. Nelson, and J. Xu, "Discrete optimization via simulation", in *Handbook of Simulation Optimization*, M. C. Fu, Ed. Springer New York, 2015, ch. 2, pp. 9–44.

[5] J. R. Swisher, S. H. Jacobson, and E. Yücesan, "Discrete-event simulation optimization using ranking, selection, and multiple comparison procedures: A survey", *ACM Transactions on Modeling and Computer Simulation*, vol. 13, no. 2, pp. 134–154, 2003.

[6] E. J. Dudewicz and S. R. Dalal, "Allocation of observations in ranking and selection with unequal variances", *Sankhyā: The Indian Journal of Statistics*, vol. 37, no. 1, pp. 28–78, 1975.

[7] Y. Rinott, "On two-stage selection procedures and related probability-inequalities", *Communications in Statistics - Theory and Methods*, vol. 7, no. 8, pp. 799–811, 1978.

[8] B. L. Nelson, J. Swann, D. Goldsman, and W. Song, "Simple procedures for selecting the best simulated system when the number of alternatives is large", *Operations Research*, vol. 49, no. 6, pp. 950–963, 2001.

[9] S.-H. Kim and B. L. Nelson, "A fully sequential procedure for indifference-zone selection in simulation", *ACM Transactions on Modeling and Computer Simulation*, vol. 11, no. 3, pp. 251–273, 2001.

[10] C.-H. Chen and L. H. Lee, *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*, ser. Stochastic Simulation Optimization. World Scientific, 2010.

[11] J. Lin, S. Teng, D. He, N. A. Pujowidianto, J. Li, S. Zhang, L. H. Lee, E. P. Chew, and C.-H. Chen, "Optimal computing budget allocation framework", in *Stochastic*

Simulation Optimization for Discrete Event Systems : Perturbation Analysis, Ordinal Optimization, and Beyond, C.-H. Chen, Q.-s. Jia, and L. H. Lee, Eds. World Scientific, 2013, ch. 9, pp. 175–202.

[12] C.-H. Chen, J. Lin, E. Yücesan, and S. E. Chick, "Simulation budget allocation for further enhancing the efficiency of ordinal optimization", *Discrete Event Dynamic Systems*, vol. 10, no. 3, pp. 251–270, 2000.

[13] C.-H. Chen, D. He, M. Fu, and L. H. Lee, "Efficient simulation budget allocation for selecting an optimal subset", *INFORMS Journal on Computing*, vol. 20, no. 4, pp. 579–595, 2008.

[14] M. C. Fu, J.-Q. Hu, C.-H. Chen, and X. Xiong, "Simulation allocation for determining the best design in the presence of correlated sampling", *INFORMS Journal on Computing*, vol. 19, no. 1, pp. 101–111, 2007.

[15] S. Gao and L. Shi, "Selecting the best simulated design with the expected opportunity cost bound", *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2785–2790, 2015.

[16] D. He, S. E. Chick, and C. H. Chen, "Opportunity cost and ocba selection procedures in ordinal optimization for a fixed number of alternative systems", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 5, pp. 951–961, 2007.

[17] S. E. Chick, "Subjective probability and bayesian methodology", in *Simulation*, S. G. Henderson and B. L. Nelson, Eds., ser. Handbooks in Operations Research and Management Science. Elsevier, 2006, vol. 13, pp. 225–257.

[18] S. E. Chick and K. Inoue, "New two-stage and sequential procedures for selecting the best simulated system", *Operations Research*, vol. 49, no. 5, pp. 732–743, 2001.

[19] P. I. Frazier and W. B. Powell, "The knowledge-gradient stopping rule for ranking and selection", in *Proceedings of the 2008 Winter Simulation Conference*, 2008, pp. 305–312.

[20] P. I. Frazier and A. M. Kazachkov, "Guessing preferences: A new approach to multi-attribute ranking and selection", in *Proceedings of the 2011 Winter Simulation Conference*, 2011, pp. 4319–4331.

[21] P. I. Frazier, J. Xie, and S. E. Chick, "Value of information methods for pairwise sampling with correlations", in *Proceedings of the 2011 Winter Simulation Conference*, 2011, pp. 3974–3986.

[22] P. I. Frazier, W. B. Powell, and S. Dayanik, "A knowledge-gradient policy for sequential information collection", *SIAM Journal on Control and Optimization*, vol. 47, no. 5, pp. 2410–2439, 2008.

[23] ——, "The knowledge-gradient policy for correlated normal beliefs", *INFORMS Journal on Computing*, vol. 21, no. 4, pp. 599–613, 2009.

[24] J. Xie, P. I. Frazier, and S. E. Chick, "Bayesian optimization via simulation with pairwise sampling and correlated prior beliefs", *Operations Research*, vol. 64, no. 2, pp. 542–559, 2016.

[25] J. Branke, S. E. Chick, and C. Schmidt, "Selecting a selection procedure", *Management Science*, vol. 53, no. 12, pp. 1916–1932, 2007.

[26] S. Andradóttir and S. Kim, "Fully sequential procedures for comparing constrained systems via simulation", *Naval Research Logistics*, vol. 57, no. 5, pp. 403–421, 2010.

[27] D. Batur, "Variance estimation in steady-state simulation, selecting the best system, and determining a set of feasible systems via simulation", PhD thesis, 2006.

[28] D. Batur and S.-H. Kim, "Finding feasible systems in the presence of constraints on multiple performance measures", *ACM Transactions on Modeling and Computer Simulation*, vol. 20, no. 3, pp. 1–26, 2010.

[29] C. M. Healey, "Advances in ranking and selection: Variance estimation and constraints", PhD thesis, 2010.

[30] C. M. Healey, S. Andradóttir, and S.-H. Kim, "Selection procedures for simulations with multiple constraints under independent and correlated sampling", *ACM Transactions on Modeling and Computer Simulation*, vol. 24, no. 3, 14:1–14:25, 2014.

[31] L. H. Lee, N. A. Pujowidianto, L. W. Li, C. H. Chen, and C. M. Yap, "Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints", *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2940–2945, 2012.

[32] S. Gao and W. Chen, "Efficient feasibility determination with multiple performance measure constraints", *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 113–122, 2017.

[33] J. Xie and P. I. Frazier, "Sequential bayes-optimal policies for multiple comparisons with a known standard", *Operations Research*, vol. 61, no. 5, pp. 1174–1189, 2013.

[34] S. R. Hunter and R. Pasupathy, "Optimal sampling laws for stochastically constrained simulation optimization on finite sets", *INFORMS Journal on Computing*, vol. 25, no. 3, pp. 527–542, 2013.

[35] R. Pasupathy, S. R. Hunter, N. A. Pujowidianto, L. H. Lee, and C.-H. Chen, "Stochastically constrained ranking and selection via score", *ACM Transactions on Modeling and Computer Simulation*, vol. 25, no. 1, 1:1–1:26, 2014.

[36] R. Szechtman and E. Yücesan, "A new perspective on feasibility determination", in *Proceedings of the 2008 Winter Simulation Conference*, 2008, pp. 273–280.

[37] ——, "A bayesian approach to feasibility determination", in *Proceedings of the 2016 Winter Simulation Conference*, 2016, pp. 782–790.

[38] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2005.

[39] E. Brochu, V. M. Cora, and N. de Freitas, *A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning*, 2010. arXiv: 1012.2599 [cs.LG].

[40] J. R. Gardner, M. J. Kusner, Z. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints", in *Proceedings of the 31st International Conference on Machine Learning*, 2014, pp. 937–945.

[41] J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani, "A general framework for constrained bayesian optimization using information-based search", *Journal of Machine Learning Research*, vol. 17, no. 160, pp. 1–53, 2016.

[42] K.-K. Cho, S.-H. Lee, and D.-S. Chung, "An automatic process-planning system for block assembly in shipbuilding", *CIRP Annals*, vol. 45, no. 1, pp. 41–44, 1996.

[43] Q. Su, "Applying case-based reasoning in assembly sequence planning", *International Journal of Production Research*, vol. 45, no. 1, pp. 29–47, 2007.

[44] Y. H. Seo, D. Sheen, and T. Kim, "Block assembly planning in shipbuilding using case-based reasoning", *Expert Systems with Applications*, vol. 32, no. 1, pp. 245–253, 2007.

[45] S. Qu, Z. Jiang, and N. Tao, "An integrated method for block assembly sequence planning in shipbuilding", *The International Journal of Advanced Manufacturing Technology*, vol. 69, no. 5, pp. 1123–1135, Nov. 2013.

[46] K. Cho, J. Oh, K. Ryu, and H. Choi, "An integrated process planning and scheduling system for block assembly in shipbuilding", *CIRP Annals*, vol. 47, no. 1, pp. 419–422, 1998.

[47] Z. Zhang, Z. Li, and Z. Jiang, "Computer-aided block assembly process planning in shipbuilding based on rule-reasoning", *Chinese Journal of Mechanical Engineering*, vol. 21, Jan. 2008.

[48] J. Kolodner, *Case-based Reasoning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[49] R. M. Marian, L. H. Luong, and K. Abhary, "A genetic algorithm for the optimisation of assembly sequences", *Computers & Industrial Engineering*, vol. 50, no. 4, pp. 503–527, 2006.

[50] Y.-J. Tseng, H.-T. Kao, and F.-Y. Huang, "Integrated assembly and disassembly sequence planning using a ga approach", *International Journal of Production Research*, vol. 48, no. 20, pp. 5991–6013, 2010.

[51] W. Zhou, J. Zheng, J. Yan, and J. Wang, "A novel hybrid algorithm for assembly sequence planning combining bacterial chemotaxis with genetic algorithm", *The International Journal of Advanced Manufacturing Technology*, vol. 52, no. 5, pp. 715–724, 2011.

[52] H.-E. Tseng, W.-P. Wang, and H.-Y. Shih, "Using memetic algorithms with guided local search to solve assembly sequence planning", *Expert Systems with Applications*, vol. 33, no. 2, pp. 451–467, 2007.

[53] L. Gao, W. Qian, X. Li, and J. Wang, "Application of memetic algorithm in assembly sequence planning", *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 9, pp. 1175–1184, 2010.

[54] J. Wang, J. Liu, and Y. Zhong, "A novel ant colony algorithm for assembly sequence planning", *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11, pp. 1137–1143, 2005.

[55] J. Yu and C. Wang, "A max–min ant colony system for assembly sequence planning", *The International Journal of Advanced Manufacturing Technology*, vol. 67, no. 9-12, pp. 2819–2835, 2013.

[56] S. Koh, C. Eom, J. Jang, and Y. Choi, "An improved spatial scheduling algorithm for block assembly shop in shipbuilding company", in *2008 3rd International Conference on Innovative Computing Information and Control*, 2008, pp. 253–253.

[57] J. G. Shin, O. H. Kwon, and C. Ryu, "Heuristic and metaheuristic spatial planning of assembly blocks with process schedules in an assembly shop using differential evolution", *Production Planning & Control*, vol. 19, no. 6, pp. 605–615, 2008.

[58] J. Zheng, Z. Jiang, Q. Chen, and Q. Liu, "Spatial scheduling algorithm minimising makespan at block assembly shop in shipbuilding", *International Journal of Production Research*, vol. 49, no. 8, pp. 2351–2371, 2011.

[59] J. Zheng, Z. Jiang, and Q. Chen, "Block spatial scheduling modelling and application in shipbuilding", *International Journal of Production Research*, vol. 50, no. 10, pp. 2744–2756, 2012.

[60] B. Kwon and G. M. Lee, "Spatial scheduling for large assembly blocks in shipbuilding", *Computers & Industrial Engineering*, vol. 89, pp. 203–212, 2015.

[61] S. Hu, S. Wang, Y. Kao, T. Ito, and X. Sun, "A branch and bound algorithm for project scheduling problem with spatial resource constraints", *Mathematical Problems in Engineering*, vol. 2015, 2015.

[62] K. J. Lee, J. K. Lee, and S. Y. Choi, "A spatial scheduling system and its application to shipbuilding: Das-curve", *Expert Systems with Applications*, vol. 10, no. 3, pp. 311–324, 1996.

[63] K.-C. Park, K.-S. Lee, S.-S. Park, and S.-H. Kim, "Modeling and solving the spatial block scheduling problem in a shipbuilding company", *Computers & Industrial Engineering*, vol. 30, no. 3, pp. 357–364, 1996.

[64] Z. Liu, D. K. H. Chua, and K. H. Wee, "Scheduling dynamic block assembly in shipbuilding through hybrid simulation and spatial optimisation", *International Journal of Production Research*, vol. 50, no. 20, pp. 5986–6004, 2012.

[65] Z. Shang, J. Gu, W. Ding, and E. A. Duodu, "Spatial scheduling optimization algorithm for block assembly in shipbuilding", *Mathematical Problems in Engineering*, vol. 2017, 2017.

[66] H.-M. Kim, J.-H. Kang, and S.-S. Park, "Scheduling of shipyard block assembly process using constraint satisfaction problem", *Asia Pacific Management Review*, vol. 7, no. 1, Mar. 2002.

[67] C. Rose and J. Coenen, "Automatic generation of a section building planning for constructing complex ships in european shipyards", *International Journal of Production Research*, vol. 54, no. 22, pp. 6848–6859, 2016.

[68] E. B. Dynkin and Y. A. A., *Controlled Markov processes*, ser. Grundlehren der mathematischen Wissenschaften. New York: Springer, 1979.

[69] D. P. Bertsekas, *Dynamic programming and optimal control*, 3rd ed., ser. Athena Scientific optimization and computation series. Belmont, MA: Athena Scientific, 2007.

[70] M. H. De Groot, *Optimal Statistical Decisions*. McGraw-Hill, 1970.

[71] H. Raiffa, *Applied statistical decision theory*, ser. Studies in managerial economics. Boston, MA: Division of Research, Graduate School of Business Administration, Harvard University, 1961.

[72] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, ser. Dover Books on Mathematics. Dover, NY: Dover Publications, 1965.

[73] M. L. Stein, *Interpolation of Spatial Data*, ser. Springer Series in Statistics. New York, NY: Springer, 1999.

[74] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization", *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[75] C. Zhu, R. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization", *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.

[76] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic gaussian process regression", in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 489–496.

[77] M. Binois, R. B. Gramacy, and M. Ludkovski, "Practical heteroscedastic gaussian process modeling for large simulation experiments", *Journal of Computational and Graphical Statistics*, vol. 27, no. 4, pp. 808–821, 2018.

[78] V. Picheny, T. Wagner, and D. Ginsbourger, "A benchmark of kriging-based infill criteria for noisy optimization", *Structural and Multidisciplinary Optimization*, vol. 48, no. 3, pp. 607–626, 2013.

[79] K. Li and K.-H. Cheng, "On three-dimensional packing", *SIAM Journal on Computing*, vol. 19, no. 5, pp. 847–867, 1990.